98ユーザーのための

CONFIG.SYS(コンフィグシス)のすべてがわかる本

池田龍之介 著 ナツメ社

98ユーザーのための

CONFIG.SYS（コンフィグ・シス）のすべてがわかる本

池田龍之介 著 ナツメ社

はじめに

「メモリ不足でソフトが起動できない」「突然、一太郎 Ver.5 が起動できなくなった」「Windows の起動が遅くて困る」などのトラブルに遭遇した経験はありませんでしょうか。これらはCONFIG.SYS というファイルにチョット手を加えることで解決することが多くあります。CONFIG.SYS は空気のように日頃は意識しない存在ですが、MS-DOS 環境を左右する重要な役割を担っています。

本書では OS は MS-DOSVer.5.0A と Windows3.1、パソコンは 32 ビットマシンを想定し、「CONFIG.SYS」の活用方法を徹底的かつ具体的に解説します。また、パソコン机の上に本書を常備していただけるように、巻末には各社のインフォメーションセンター一覧などの便利なデータも添付しています。

本書は次の構成で記述されています。

●基礎知識編

CONFIG.SYS とは何か、メモリとは何かなど、パソコン環境を考える上で基本となる事柄の概念や仕組みをわかりやすく図解・解説します。ビギナーユーザーは、ここから読み始めてください。腕に覚えのあるユーザーは、ここを飛ばして徹底活用編へ進んでください。

●徹底活用編

いかにメモリのフリーエリアを広くするか、いかにディスクアクセスを高速にするかなど、やや高度な設定テクニックを紹介します。メーカーでは絶対に教えないトリッキーな設定方法も紹介します。

●リファレンス編

CONFIG.SYS コマンド・各社メモリドライバなど、CONFIG.SYS をカスタマイズする上で不可欠なコマンド、ドライバの機能リファレンスです。マニュアルに記載されてない「隠しオプション」などについても紹介します。

最後に本書執筆に当たって、ナツメ出版企画株式会社の甲斐氏・山縣氏をはじめ、裏方で支えてくれたすべてのスタッフに心から感謝申し上げます。そして本書が快適なパソコン環境を構築する上で、少しでも読者のお役に立てればこれ以上の幸せはありません。

1994 年初秋　　　　池田　龍之介

☆ 動作確認を行ったシステムの基本構成

パソコン本体	：メインマシン PC-9821As/U2 (NEC)、JBOXODP486DX33 (Intel) ：サブマシン PC-9801DA/U2、PC-9801SX/20、PC-9801SX/T (NEC)
増設メモリ	：12MB
ディスプレイ	：PC-KM171 (NEC)
ハードディスク	：500MB
プリンタ	：LBP-A404GII (Canon)
基本ソフト	：MS-DOS Ver.5.0A、Windows Ver.3.1 (NEC/Microsoft)

☆ 使用するアテンションマーク

注意	特に注意が必要な操作です
重要	ポイントとなる重要な知識です
高度	やや高度な操作・知識です （ビギナーは読み飛ばしてもかまいません）
参照	参照すべきページ番号を示します

目　次

基礎知識編

徹底活用編

第1章 MS-DOS Ver.5.0A を徹底活用する

第2章 MemoryServerⅡを徹底活用する

第3章 MELWAREを徹底活用する

第4章 主要ソフトの快適環境を設定する

リファレンス編

第1章 CONFIG コマンド一覧

第2章 MS-DOS デバイスドライバ一覧

第4章 トラブル対策一覧

基礎知識編

ここではパソコン環境を決定づける
摩訶不思議なファイル
「CONFIG.SYS」を
征服するための前提となる
基礎的な知識を身に付けます

第1章
CONFIG.SYSの基礎知識

そもそもCONFIG.SYSとは何者なのでしょうか。このファイルを知るとMS-DOSが見えてきます
ここでは、CONFIG.SYSに関する基礎的な知識について解説します

まずMS-DOSの正体を知る

CONFIG.SYSについて説明する前に、まずMS-DOSの概要を理解しましょう。

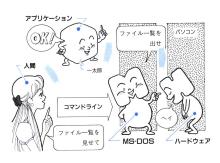
そもそもエムエス・ドスって何だ?

MS-DOS（エムエス・ドス）とは、MicroSoft Disk Operating Systemの頭文字をとって名付けられたOSです。OS（オーエス=Operating System）とは、パソコン（ハードウェア）とアプリケーション（一太郎やLotus1-2-3など）との間に入り、双方の「橋渡し役」をするソフトウェアのことです。

たとえばパソコン（ハードウェア）がタクシーだとすると、MS-DOS（OS）は運転手で、アプリケーションはお客です。お客が「東京駅まで」と言えば、後は運転手が万事やってくれます。お客が「最初はギアをローに入れてからアクセルをゆっくりふかしてください」「信号が赤だからブレーキペダルを踏んでください」などという細かい指示を逐一出す必要はないのです。

☆OSはアプリケーションとパソコンの仲介役をする

これと同じく、アプリケーションから「Bドライブのディスクに入っているファイル名の一覧を表示せよ！」という命令（コマンド）を受けると、後はMS-DOSがパソコンを動かしてファイル名を受け取って、それをアプリケーションに知らせます。ディスク上のどの位置から、どういう手順でファイル情報を読み込むかなどという操作・判断は、基本的にはMS-DOSにまかせておけばよい仕組みになっています。

もし、OSの役割をアプリケーションが独自に行ってしまったら、プログラムの規模や開発工数は限りなく巨大化し効率が悪くなります。ハードウェアとの直接的な作業をOSというソフトウェアにまかせることによって、アプリケーションは本来の業務に専念できるわけです。このようにOSは、ハードウェアとアプリケーションとの仲介役をこなす「縁の下の力持ち」的な存在なのです。

メーカーごとに異なるMS-DOS

実際にパソコンショップでMS-DOSを購入しようとすると、「NEC製MS-DOS」「IBM製MS-DOS」など、マイクロソフト社のオリジナルMS-DOSのほかに、パソコンメーカーごとにMS-DOSがあります。それは、メーカーによってパソコンのハードウェアの基本設計が異なるため（これを「互換性がない」と言う）、各々の仕様に合わせてMS-DOSに改良を加えるからです。

また最近では、NEC以外のメーカー各社からPC/AT互換機（通称DOS/Vマシン）が発売されています。これはハードウェアの基本設計がすべて同じであることになっていますが、ここでもメーカーごとに微妙な差異がある場合があります。したがって、マシンにバンドル（添付）されているMS-DOSを使用するよう製品の許諾契約書にあるのが一般的です。すなわち、A社のDOS/Vマシンに添付のMS-DOSが、B社のDOS/Vマシンで正常に動くとは限らないのです。また、DOS/VマシンではPC-DOSやDR-DOSといったMS-DOS互換のOSがあり、その選択肢が多いことも特徴の1つです。このようなことから、新しくMS-DOSを購入する際は、パソコンメーカーに相談してから行った方が安全です。

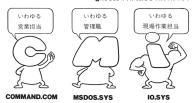
Column

●MS-DOS誕生の歴史

1980年、米国IBMはパソコン市場への参入第一弾として、「IBM-PC」という16ビットマシンを極秘開発していました。このパソコンを生かすためのOSの必要性を感じたIBMは、MS-BASICの開発で有名だったマイクロソフト社のビル・ゲイツ氏（現会長）に新OSの開発を依頼しました。ちなみに当時のマイクロソフト社は社員数人のベンチャー企業でした。当時はパソコンOSと言えばCP/M（デジタルリサーチ社）が圧倒的な主流でしたが、デジタルリサーチ社とIBMの交渉がうまくいかずに、最終的にマイクロソフト社が独自にOSを作ることに落ちつきました。OS開発のノウハウがなかったマイクロソフト社は、急遽、シアトル・コンピュータ・プロダクツ社が販売していた「QDOS」を5万ドルで買い取り、IBM-PC用に改良しました。そして1981年8月、IBM-PC出荷と同時に「MS-DOS」が世に出たわけです。その後、IBM-PCがパソコン市場を制覇し、それと同時にMS-DOSもパソコンOSの代名詞ともいうべき存在になりました。

MS-DOS を構成する３つのファイル

MS-DOS のシステムディスクには多くのファイルが格納されています。しかし、MS-DOS 本体は次の３つのファイルから成り立っています。最低限この３つがないと MS-DOS は起動できません。

☆MS-DOS の本体である３つのシステムファイル

■COMMAND.COM

COMMAND.COM は依頼元（キーボードやアプリケーション）から受け取った命令を解析し、MSDOS.SYS へ伝達する仕事をします。またその処理結果を受け取って、依頼元へ返します。コマンドプロンプト「A>」やエラーメッセージの表示なども COMMAND.COM の役目です。

■MSDOS.SYS

MSDOS.SYS は MS-DOS の根幹をなすカーネル（核）プログラムで、MS-DOS の諸作業を監視・制御します。主な役割はディスク管理/ファイル管理/メモリ管理/デバイス管理などです。たとえば COMMAND.COM から送られる処理依頼を解析し、IO.SYS の中から必要な実行ルーチンを呼び出すなどの作業を行います。

■IO.SYS

IO.SYS はデータ入出力を行うための各実行ルーチンを寄せ集めたプログラムです。キーボード/プリンタ/ディスク/RS-232C/ディスプレイなどの周辺機器とのデータ入出力を実際に行います。正確には IO.SYS はパソコンに内蔵されている BIOS（バイオス）というプログラムに対して命令を発しています。

CONFIG.SYS ファイルの役割を知る

MS-DOS を快適に走らせるには、CONFIG.SYS を征服しなければなりません。ここでは CONFIG.SYS の機能や仕組みについて解説します。

CONFIG.SYS って何だ?

CONFIG.SYS (コンフィグ・シスと読む) の「CONFIG」とは「Configuration =配置、地形、輪郭」といった意味があります。すなわち CONFIG.SYS は「MS-DOS の輪郭を設定するファイル」です。MS-DOS は起動時に 1 回だけこのファイルを読み、その記述内容に従って MS-DOS の動作環境を自動的に構築します。

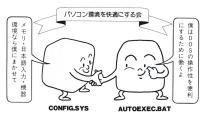
CONFIG.SYS は MS-DOS 環境を決定づける

たとえば、CONFIG.SYS が存在しないドライブから MS-DOS を起動することもできます。この場合は必要最小限の機能しかない小さな MS-DOS 環境

が構築されます。この環境では一太郎やLotus1-2-3などの大きなプログラムを起動することはできません。また日本語を入力することもできません。拡張メモリを活用したり、日本語入力ができる環境にするには、それらをMS-DOSに組み込むための命令語（CONFIGコマンド）をCONFIG.SYSに記述しなければなりません。

例えるなら、MS-DOSそのものは自動車のエンジン部分に相当します。そのエンジンからどういう車種を作るかは、ユーザーがCONFIG.SYSで自在にデザインできるわけです。逆に言うとCONFIG.SYSを知らないと、せっかくのマシン能力が眠ったままになるという無駄が生じてしまいます。このようにユーザーの目的・機器環境に応じて、自由にMS-DOSを改築するための指示書が「CONFIG.SYS」の役割です。

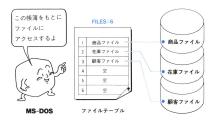
■もう1つの環境ファイル「AUTOEXEC.BAT」

CONFIG.SYSと並んでMS-DOS起動時に1回だけ実行されるファイルに「AUTOEXEC.BAT（オートイグゼック・バット=自動実行ファイル）」があります。MS-DOSの環境構築を考えるときには、CONFIG.SYSとAUTOEXEC.BATはなくてはならないファイルです。この2つのファイルはいずれも、ユーザーが自由に作成・編集できるテキストファイルです。AUTOEXEC.BATにはコマンドパスの設定（PATHコマンド）・環境変数の定義（SETコマンド）・TSR（常駐型プログラム）の実行など、起動時に1回だけ実行すればよいコマンドを登録しておきます。CONFIG.SYSもAUTOEXEC.BATも、起動ドライブのルートディレクトリに置く必要があります。

CONFIG.SYS はこんな仕事をする

次に示すCONFIG.SYSの内容を見ながら、おおまかな役割を見てみましょう。

```
FILES=20
BUFFERS=20
SHELL=A:¥COMMAND.COM A:¥ /P /E:384
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=65 /UMB /MOVEHDBIOS
DEVICE=A:¥DOS¥SMARTDRV.SYS 128 /E
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
DOS=HIGH, UMB
```

■FILES ：同時に扱えるファイル数を指定する p.192

MS-DOS は起動時に FILES で指定した値だけ、メモリ上にファイルテーブル（ファイルを管理するための帳簿）を作ります。ファイルを読み書きする際には、このファイルテーブルをたよりに行います。たとえば「FILES=10」と設定すると、10 個までのファイルを同時に扱えるようになります。Windows 3.1 を起動するには最低でも「FILES=30」を指定しなければなりません。

■BUFFERS ：ディスクバッファの数を指定する

p.36、185

パソコンはハードディスクやフロッピーディスクにアクセス（読み書き）する際に、入出力バッファと呼ばれる領域（コンベンショナルメモリ上に作成される）に、一時的にデータを溜めておきます。ディスクへのアクセスは必ず入出力バッファを経由して行われます。

一度読み込んだデータは入出力バッファに溜まっています。次にデータ読込命令があったときに入出力バッファ内に目的のデータがあれば、それをCPUに渡します。したがって、ディスクバッファの容量が大きければ、ディスクへのアクセス回数が少なくなり、結果的に高速にパソコンが動作します。

■SHELL ：シェルプログラムを指定する

p.196

SHELLコマンドはシェルプログラム（またはコマンドプロセッサと呼ぶ）のファイル名を指定します。シェルプログラムは入力したコマンドを受け付けたり、エラーメッセージを表示したりします。現在ではシェルプログラムは「COMMAND.COM」しか使われていません。

CONFIG.SYSにSHELLコマンドを指定しないと、自動的に起動ドライブのルートディレクトリにあるCOMMAND.COMが、シェルプログラムとして読み込まれます。したがって、SHELLコマンドは特に指定する必要がないように見えるかもしれません。ところが、そうはいかないのです。

COMMAND.COMはコンベンショナルメモリ（メインメモリ）上では、常駐部（常に記憶されている部分）と非常駐部（一時的に記憶されている部分）の2つに分けて配置されます。ワープロや表計算などの大きなプログラムを実行する際にメモリが不足すると、この非常駐部の上から上書きして使います。こ

れはメモリ不足を補うために考え出された苦肉の策です。

このときCOMMAND.COMの非常駐部は破壊された状態になっています。プログラムを終了して、コマンドプロンプト(A>やA:¥>など)を表示するには、再びCOMMAND.COMをディスクからメモリに読み込む必要があります。このときSHELLコマンドで指定してあるパスから、COMMAND.COMが読み込まれるわけです。この指定がないと、「COMMAND.COMがみつかりません」というエラーが表示され、処理を続行できなくなる場合があります。

■DEVICE :MS-DOSに特定の機能を付加する

p.187

DEVICEコマンドはMS-DOSにデバイスドライバ(周辺機器や日本語入力プログラムなどを制御するためのプログラム)を付加します。たとえば次のデバイス指定では、プロテクトメモリ(**p.38**)をMS-DOSで使用できるようにしています。

```
DEVICE=A:¥DOS¥HIMEM.SYS                      ←——— XMSドライバ
DEVICE=A:¥DOS¥EMM386.EXE /P=65 /UMB /MOVEHDBIOS ←— EMSドライバ
DEVICE=A:¥DOS¥SMARTDRV.SYS 128 /E            ←——— ディスクキャッシュドライバ
```

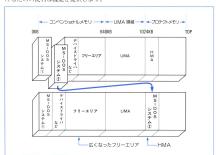
また次のデバイス指定では、日本語入力プログラム「ATOK8」を使用できるようにしています。

このようにDEVICEコマンドによって、MS-DOSに目的の機能を次々に付加していくことができます。CONFIGコマンドの中で最も重要な地位を占めるコマンドです。

■DOS ：XMSメモリの用途を指定する *p.190*

「DOS=HIGH」という記述は、MS-DOSシステムの一部分をHMA（ハイメモリ）へ移動する命令です。これによってコンベンショナルメモリに常駐するはずだった約59KBのプログラムがHMA（***p.50***）へ移動します。その分だけコンベンショナルメモリのフリーエリア（空き領域）は増えます。

また「DOS=UMB」という記述は、DEVICE行で指定しているEMM386.EXEで確保したUMBメモリ（***p.49***）の使用を許可する機能があります。これによってUMBが実際に使えるようになります。

このようにDOSコマンドは、コンベンショナルメモリのフリーエリアを広げるための便利な機能を提供します。

■CONFIG.SYSとメモリの関係

CONFIGコマンドがMS-DOSを高機能にすることは、これまで述べてきた通りです。しかし、ここで注意しなければならないのが、「メモリとの関係」です。FILES値/BUFFERS値を増やすごとにコンベンショナルメモリのフリーエリアは小さくなります。またデバイスドライバを1個付加しただけで、相当のメモリが消費されます。結局、限りあるメモリ資源の中でいかに高機能を実現するかが、CONFIG.SYS設定のキーとなります。一言でいうと、CONFIG.SYS設定の奥義はメモリ設定にあると言えるでしょう。詳しくは「徹底活用編」以降を参照してください。

MS-DOS 起動のメカニズムを知る

MS-DOSは起動時にどのようなタイミングでCONFIG.SYSやAUTOEXEC.BATを読み込むのでしょうか。MS-DOSの起動メカニズムを見てみましょう。

MS-DOS 起動時の画面

```
NEC PC-9800 ｼﾘｰｽﾞ ﾊﾟｰｿﾅﾙ ｺﾝﾋﾟｭｰﾀ

ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 5.00A-H
Copyright (C) 1981,1992 Microsoft Corp. / NEC Corporation

VMM386 Virtual Memory Server Release 3.02
Copyright (C) 1992-1994 I･O DATA DEVICE Inc. All rights reserved.

EMS / XMS / VCPI として 14016 KBytes 使用可能です.
UMB が 91 KBytes 使用可能です.

日本語変換システムATOK8 Ver.1.0 /R.1 for PC-9800
(C)1993 株式会社ジャストシステム
ATOKをEMSメモリへ組み込みました.

ATOK8拡張ドライバ Ver.1.00
(C)1993 株式会社ジャストシステム

Command ﾊﾞｰｼﾞｮﾝ 5.00A
A:¥>
```

操作

①電源 ON

パソコンの電源スイッチを ON にします。

②初期化ルーチンの起動

- ROM BIOS の中の初期化ルーチン（起動用プログラム）が起動します。
- 装着されているメモリをチェックします。
- 接続されている周辺機器を調べます。SCSI ハードディスクが接続されている場合は、使用する ID 番号のチェックを行います。
- ハードディスクの BPB（ディスクの先頭にあるディスク情報エリアで、セクタ数・シリンダ数などが格納されている）を読み込みます。

③IPL の実行 p.17

- ブートストラップローダが起動し、IPL（Initial Program Loader=CPU に OS を実行させるためのプログラム）を読み込みます。
- IPL は接続されているディスクから MSDOS.SYS と IO.SYS を探しにいきます。通常は FDD 装置 1→FDD 装置 2→HDD 装置の順に検索します。た

だし起動ドライブがあらかじめ指定されている場合は、それを優先します。
・MSDOS.SYS と IO.SYS を読み込みメモリに配置します。

④CONFIG.SYS の読み込み

IO.SYS は起動ドライブのルートディレクトリから CONFIG.SYS を探します。発見すると CONFIG.SYS の先頭行から解析し、記述内容に従って MS-DOS の環境を設定します。CONFIG.SYS で指定した設定内容は MS-DOS システムに近いレベルで構築されるため、アプリケーション側から変更することはできません。設定内容は、リセットボタンが押されるか電源 OFF まで保持されます。

⑤コマンドプロセッサの読み込み

IO.SYSはCONFIG.SYSで指定されたシェルプログラム(通常はCOMMAND.COM)を読み込み、コンベンショナルメモリに配置します(ここからの制御は COMMAND.COM に移されます)。

⑥AUTOEXEC.BAT の実行

COMMAND.COM は起動ドライブのルートディレクトリから AUTOEXEC.BAT を探し、発見すると実行します。存在しなければ DATE と TIME の 2 つのコマンドを実行し、コマンドプロンプト (A> や A:¥> など) を表示します。

Column

●Windows95 で CONFIG.SYS はどうなる?

Windows95 (Windows3.1 の次期バージョン) では、MS-DOS を経由せずに、Windows がダイレクトに起動します。したがって CONFIG.SYS と AUTOEXEC.BAT は、基本的には必要としません。ただし、ユーザーが独自に CONFIG.SYS や AUTOEXEC.BAT を作成することはできます。これらのファイルを発見すると、Windows95 はその内容を読み込んでシステム全体に反映させます。

将来的に登場するであろうサードパーティ製の強力なメモリマネージャを活用したり、MS-DOS アプリケーションを Windows95 で動かす場合は、やはり CONFIG.SYS や AUTOEXEC.BAT に関する知識は必要になると思われます。

CONFIG.SYS を作成する

ここでは実際に CONFIG.SYS を作成・修正する方法について説明します。

現在の CONFIG.SYS 内容を確認する

現在の CONFIG.SYS の内容を画面に表示しましょう。

①TYPE コマンドの実行

MS-DOS コマンドラインから次のように入力します。

```
TYPE A:¥CONFIG.SYS ⏎
```

☆CONFIG.SYS の中身を覗いてみる

```
A:¥ TYPE A:¥CONFIG.SYS   ←―――― 入力するコマンド
DEVICE=A:¥UTLTY¥HSB.EXE  VC Y2 TD
BUFFERS=25
FILES=80
FCBS=1
SHELL=A:¥COMMAND.COM /P /E:272
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE  /M /U=D0-D7,DE-DF,E8-F4 /W=CC /NECID
DEVICE=A:¥MDEV¥IOSPRO¥DC10.EXE 3072 /W=2048
DEVICE=A:¥DISKX2¥DISKX.SYS /UMB B:0
DEVICE=A:¥DISKX2¥XDRV.COM /IU B:0
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=B:¥ATOK8¥ATOK8A.SYS /UCF=B:¥ATOK8¥ATOK8.UCF
DEVICE=B:¥ATOK8¥ATOK8B.SYS
DEVICE=B:¥ATOK8¥ATOK8EX.SYS
DOS=HIGH,UMB
LASTDRIVE = Z
DEVICEHIGH = B:¥ICMWIN¥CDROM.SYS /D:CD500E
DEVICE=A:¥UTLTY¥HSB.EXE  TS

A:¥>
```

TYPE コマンドは MS-DOS の内部コマンド（MS-DOS に内蔵されたコマンド）なので、コマンドプロンプト（A> や B>）が表示されていれば、常に実行できます。CONFIG.SYS は必ず起動ドライブのルートディレクトリに存在します。

エディタでCONFIG.SYSを編集する

現在、「A:¥DOS」にはマウスドライバ「MOUSE.SYS」が格納されています。このマウスドライバが使えるように、CONFIG.SYSに登録しましょう。なお、ここではMS-DOS Ver.5.0に付属するSEDITコマンドを使います。

MOUSE.SYSをCONFIG.SYSに登録する

```
A:¥CONFIG.SYS                    <挿入>   14/  19行 24桁  空メモリ: 98  [HELP]:説明
DEVICE=A:¥UTLTY¥HSB.EXE  VC Y2 TD↓
BUFFERS=25↓
FILES=80↓
FCBS=1↓
SHELL=A:¥COMMAND.COM /P /E:272↓
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE  /M /U=D0-D7,DE-DF,E8-F4 /W=CC /NECID↓
DEVICE=A:¥MDEV¥IOSPRO¥DC10.EXE 3072 /W=2048↓
DEVICE=A:¥DISKX2¥DISKX.SYS /UMB B:0↓
DEVICE=A:¥DISKX2¥XDRV.COM /IU B:0↓
DEVICE=A:¥DOS¥KKCFUNC.SYS↓
DEVICE=B:¥ATOK8¥ATOK8A.SYS /UCF=B:¥ATOK8¥ATOK8.UCF↓
DEVICE=B:¥ATOK8¥ATOK8B.SYS↓
DEVICE=B:¥ATOK8¥ATOK8EX.SYS↓
DEVICE=A:¥DOS¥MOUSE.SYS↓   ←―― 挿入する行
DOS=HIGH,UMB↓
LASTDRIVE = Z↓
DEVICEHIGH = B:¥ICMWIN¥CDROM.SYS /D:CD500E↓
DEVICE=A:¥UTLTY¥HSB.EXE  TS↓
[EOF]
```

操　作

①エディタの起動

MS-DOSコマンドラインから次のように入力します。

SEDIT A:¥CONFIG.SYS ⏎

②MOUSEドライバの記述 p.209

・MOUSEドライバを記述したい行（ここでは14行目）の先頭にマウスカーソルを移動し、⏎キーを押します（空白行の挿入）。

・挿入された空白行に、次の1行を記述します。

DEVICE=A:¥DOS¥MOUSE.SYS ⏎

③変更内容の保存

・f·10（終了）キーを押します。

・「文書を保存しエディタを終了」にカーソルを移動し、⏎キーを押します。

解　説

■作成したCONFIG.SYSの反映

p.25

MS-DOSがCONFIG.SYSを読み込むのは起動時の1回だけです。したがって、作成・修正したCONFIG.SYSをMS-DOSシステムに反映させるには、CONFIG.SYSを保存した後にリセットボタンを押してください（再起動の実行）。

■CONFIG.SYSを作成する3つの方法

CONFIG.SYSを作成編集する主な方法には、およそ次の3つがあります。

エディタによる作成

テキストエディタは本来はプログラムの編集用に開発されたユーティリティですが、現在では広くテキストファイル作成に使われています。ポピュラーなテキストエディタには、MIFES（メガソフト）やVzエディタ（ビレッジセンター）があります。

CUSTOMコマンドによる作成

MS-DOSに付属するCUSTOMコマンドで、対話式に作成できます。

アプリケーションによる作成

メモリマネージャ（MemoryServer IIやMELWARE）やアプリケーション（一太郎Ver.5など）をインストールすると、CONFIG.SYSとAUTOEXEC.BATは自動的に生成されます。

ただし、CUSTOMコマンドやアプリケーションによって作成するCONFIG.SYSは、必ずしも最適な内容になるとは限りません。最適なCONFIG.SYSを作るには、最終的にはエディタによる修正が必要です。

エディタなしで即座にCONFIG.SYSを作る

現在、エディタのない環境でMS-DOSが起動しています。次のような内容のCONFIG.SYSをすばやく作成しましょう。

```
BUFFERS=20
FILES=20
FCBS=1
SHELL=¥COMMAND.COM /P
```

操　作

①COPYコマンドの実行

MS-DOSコマンドラインから次のように入力します。

```
COPY CON A:¥CONFIG.SYS ↵
```

②CONFIG.SYSの内容記述

上図のような内容を入力します（各行末で ↵ キーを押します）。

③ファイルの保存

- 4行目の行末で CTRL キーを押しながら Z キーを押します（EOFコードの挿入）。
- ↵ キーを押します。これで「1個のファイルをコピーしました」と表示され、CONFIG.SYSが作成されます。

解　説

■標準デバイスの活用

「CON」とは、MS-DOSが標準で用意するデバイスドライバの名称です。CONはキーボードや画面を制御するためのもので、上記のように「COPY CON CONFIG.SYS ↵」と入力することで、キーボードからCONFIG.SYSを作成できます。入力を終了するにはファイルの終端を表すEOF（End Of File）コード（ CTRL ＋ Z を押す）を入力します。緊急にCONFIG.SYSやAUTOEXEC.BATを作成する場合に、覚えておくと便利なテクニックです。

第２章

メモリの基礎知識

MS-DOS や Windows のアプリケーションを快適に動かすには、CONFIG.SYS におけるメモリの設定がポイントになります
ここでは、MS-DOS や Windows でのメモリ活用の基礎知識について解説します

メモリの「いろは」を知る

まずメモリの概念・役割・種類について理解しましょう。

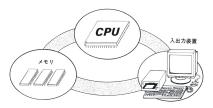
メモリって何だ？

パソコンを扱うときに「メモリが足りない」とか「メモリを増設しなければ」などという言葉をよく聞きます。この「メモリ」とは一体何者で、どんな役割を果たしているのでしょうか。メモリには文字どおり「Memory＝記憶する」という意味があります。コンピュータの世界では広い意味では文字や数値などのデータを記憶しておく場所（ハードディスクなども含めて）をメモリと呼びます。また、狭い意味では半導体メモリを指します。本書も含め、一般的には「メモリ」というと半導体メモリを指すと考えてください。

このメモリの役割を理解するには、パソコンのハードウェア内部の機構をある程度理解しておく必要があります。パソコンは大ざっぱに言うと、CPU・メモリ・入出力装置という３つの核から成り立っています。

☆パソコンを構成する３つの要素

CPU（中央演算装置）

p.39

CPU はプログラムの実行/演算/データの入出力を制御するパソコンの頭脳部分です。MS-DOS や Windows を動かすパソコンでは、米国のインテル社の開発した CPU チップ（i80386/i80486/Pentium など）、またはその互換チップが使われています。

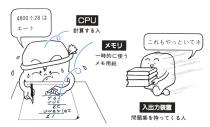
入出力装置

入出力装置はフロッピーディスク装置/キーボード/ディスプレイのように外部からデータや命令を入力したり、出力したりする装置です。

メモリ

メモリは CPU の処理を助けるために使う一時的な作業領域です。CPU が計算するときに一時的に使う「メモ用紙」のようなものと考えてよいでしょう。CPU が処理を行う過程で、ディスクから読み込んだプログラムやデータを格納したり、演算途中の作業情報を書き込んだりします。

ハードディスクやフロッピーディスクはデータをアクセスするためには、円盤（データが記録してある媒体）をモーターで回転させたり、ヘッド（データを読み取るセンサー）を移動したりという物理的な動作が必要です。しかし、メモリは半導体に電気的にアクセスするため、物理的な動作を伴いません。したがって極めて高速に CPU とのデータのやりとりができます。

メモリは CPU の処理を助ける

ハードウェア的に見たメモリの使われ方

メモリにはROM (Read Only Memory=読み出し専用メモリ) とRAM (Random Access Memory=書き出し/読み出しメモリ) の2種類があります。ROMはあらかじめ書き込まれた情報を読み取ることしかできないタイプのメモリです。RAMは文字どおり情報の登録や読み出しが可能なタイプのメモリです。これらのメモリはその特性に応じて、パソコン内部で次の3つの用途に使われています。

■システムBIOSなどの格納メモリ(ROM)

PC-98シリーズ (以下98シリーズ) では、N88-BASIC/BIOS/日本語文字パターンなどがROMに記録されています。これらの情報はパソコン製造時にROMに記録するので、後からユーザーが自由に情報を登録・修正することはできません。また電源をOFFにしても内容は保持されます。98シリーズではN88-BASICとBIOSを格納するROMが96KBあります。98NOTEではノートメニューが含まれるので、128KBのROMが搭載されています。日本語文字パターンやその他のROM容量については公開されていません。ROMからの情報の取り出しはOSまたはアプリケーションが行うので、通常はユーザーが直接アクセスすることはできません。

■画面表示情報の格納メモリ(VRAM)

画面に文字や図形を表示するための情報を記録するメモリをVRAM (Video RAM) と言います。VRAMはRAMの一種で、文字を表示するためのテキストVRAM (TVRAM)、グラフィックを表示するグラフィックVRAM (GVRAM) の2種類があります。98シリーズではTVRAMに12KB、GVRAMに256KB (98MATE以降の機種では512KB) を実装しています。

MS-DOS上では文字の表示にTVRAMを使いますが、Windows上では文字の表示にはGVRAMを使います。これによりフォントさえ揃えれば、文字サイズ・書体を自在に表示できるようになっています。

■アプリケーション作業用のメモリ(RAM)

p.36

パソコンのカタログなどで「ユーザーズメモリ　標準5.6MB 最大37.6MB」

などと記載されているメモリは、アプリケーションやユーザーが自由に使えるRAMです。メインメモリ/EMSメモリ/UMBメモリなどという呼び名は、このRAMをMS-DOSがソフトウェア的に区分けするときの定義名です。本書の「パソコンの快適環境を設定する」というテーマでは、もっぱら、このRAMの設定方法が中心になります。このメモリを増設することで、Windowsや一太郎を高速に動作させることができるようになります。

カタログに記載されるメモリ

		PC-9821 Ce2 modelS2	PC-9821 Ce2 modelS2D	PC-9821 Ce2 modelS1
		Windows3.1インストールモデル		FDDモデル
CPU		i486™SX(25MHz)		
メモリ	ユーザーズメモリ	標準5.6MB/最大37.6MB *1		標準1.6MB/最大33.6MB *1
メモリ	ビデオRAM	512KB(グラフィックVRAM)、12KB(テキストVRAM)		
表示機能	テキスト表示	英数カナ:80文字×25行/80文字×20行、漢字(16ドット、ゴシック体):40文字×25行/40文字×20行 テキストVRAMはローカルバス経由でCPUからの直接アクセスが可能		
表示機能	グラフィック表示	解像度(表示色):640×400ドット(4,096色中16色または1,677万色中256色)2画面 640×480ドット(1,677万色中256色)1画面 グラフィックVRAMはローカルバス経由でCPUからの直接アクセスが可能		
内蔵補助記憶装置	フロッピィディスク	3.5インチFDD(1.44MB/1MB/640KB対応)×1 増設用3.5インチFDD(2台目FDD)〈別売〉の追加増設が可能	3.5インチFDD(1.44MB/1MB/640KB対応)×2	
内蔵補助記憶装置	固定ディスク	170MB、ライトキャッシュ対応		なし(オプション)
内蔵補助記憶装置	CD-ROM	倍速(平均アクセスタイム330ms〈1/3ストローク〉)、データ転送速度300KB/S、フォトCD(マルチセッション)対応		
汎用拡張スロット		3スロット		
サウンド機能		PCM録再機能2チャンネル(ステレオ:量子化8ビット/16ビット、サンプリングレート4.13KHz〜44.1KHz)、 FM音源6和音(ステレオ)、リズム音源6和 専用マイクロホン(PC-9821A-U0		
	ディスプレイ	アナログRGB×1		
	入力	マイクロホン入力		

Column

●SRAMとDRAM

RAMはそのデータ保持の方式によって、ダイナミックRAM(DRAM)とスタティックRAM(SRAM)の2種類に分けられます。DRAMは一定時間ごとにリフレッシュ(再書き込み)しないと、記憶内容が消えてしまうため、結果的に消費電力が大きくなるという欠点があります。しかし、SRAMに比べて価格が圧倒的に安いため、多くのパソコンはDRAMを使用しています。

しかし、現在ではこの普及型DRAMは速度向上のための設計変更が迫られています。かつてはパソコンのカタログには「ノーウェイト」と表記するのが流行だったことがあります。これはCPUがメモリをアクセスする際に、待ち時間がない(つまり高速にアクセスできる)ことを表します。しかし最近のCPUの高速化に伴い、むしろ現在のDRAMはCPU処理の足を引っ張ってしまい、「ウェイト」をかけざるを得なくなっているのが実状です。今後はより高速な次世代式DRAM(EDRAM/CDRAM/RDRAMなど)が登場してくると思われます。

MS-DOS が定義するメモリ

MS-DOS を起動すると、MS-DOS はパソコンに搭載されているメモリを、次のように区画整理して使用します。

MS-DOS が管理するメモリ

■コンベンショナルメモリ(Conventional Memory)

MS-DOS はもともと 8086 系 CPU を使うための OS として作られたので、1024KB (1MB) までのメモリを管理できるようになっています。この 1MB のメモリの内、最初の 640KB をコンベンショナルメモリと言います。主記憶メモリ/メインメモリ/ユーザーズメモリなどと呼ばれることもありますが、本書では「コンベンショナルメモリ」に統一します。

MS-DOS の起動時、コンベンショナルメモリには MS-DOS システム・割り込みベクタ (ソフトウェア割り込みの帳簿のようなもの)・CONFIG.SYS で指定したデバイスドライバ、などが配置されます。

640KB からそれらのメモリを差し引いたフリーエリア (空き領域) は、MS-DOS コマンドや MS-DOS 対応プログラムを実行するときに、プログラムの一部分を読み込むために使われます。ただし、このフリーエリアがプログラムを読み込めるだけの容量がない場合は、「メモリ不足です」「メモリが足りません」などというエラーメッセージが表示され、アプリケーションが実行できなくなります。したがって、できるだけフリーエリアが広くなるように CONFIG.SYS を設定するのが、腕の見せ所になるわけです。

MS-DOS 起動時に占有される領域 / 実行するプログラムを読み込む領域 / UMA 領域

0KB　640KB

割込みベクタテーブル | MS-DOSシステム | デバイスドライバ | COMMAND・COM | 常駐型プログラム | COMMAND・COM 非常駐部 | フリーエリア

UMA 領域(Upper Memory Area)

MS-DOS が管理する 1MB のメモリの内、コンベンショナルメモリに続く 384KB のメモリ空間を UMA 領域と言います。UMA 領域はもっぱらハードウェアが使う ROM 空間で、ROM BASIC・ROM BIOS（入出力制御プログラム）・EMS ボード・サウンドボード・ハードディスクなどを制御するのに使われます。通常はユーザーが自由にアクセスできません。UMA 領域はシステム領域や拡張 ROM 領域などと呼ばれることもありますが、本書では「UMA 領域」に統一します。

UMA 領域が実際にどのように使われているかは、使用するパソコン機種や接続する周辺機器の種類・数によって異なります。また UMA 領域は必ずしもその全部が使われているわけではなく、未使用領域が数十 KB~128KB ほどあります。32 ビットマシン（i80386 以上）であれば、プロテクメモリの一部を貼り付けて特殊な RAM 空間として使うこともできます。これを UMB(Upper Memory Block) と言います。UMB 領域にはデバイスドライバや常駐型プログラムを格納することができ、これによってコンベンショナルメモリのフリーエリアを広くすることができるようになります。

■プロテクトメモリ

MS-DOS が管理できる 1MB（コンベンショナルメモリ＋UMA 領域）に続くメモリ領域を「プロテクトメモリ」と言います。パソコンのカタログで「ユーザーズメモリ」「主記憶 RAM」などと記載されている RAM は、コンベンショナルメモリとプロテクトメモリを合わせた容量が記載されています。

プロテクトメモリは 32 ビット CPU（i80386 以上）だけがアクセスできる拡張メモリ空間です。1MB という MS-DOS のメモリ管理の限界を解消するために、32 ビット CPU 固有の機能を利用して、プロテクトメモリをアクセスします。XMS 規格や DPMI 規格など、プロテクトメモリの使い方には数種類があります（詳しくは ***p.48*** 以降を参照）。

Column

●メモリを数える単位

コンピュータの世界ではデータの大きさ（容量）は、次の単位で数えます。

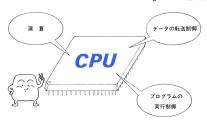
単位	読み	意　味
bit	ビット	データの最小単位(1ビットは2進数1桁に当る)
Byte(B)	バイト	1バイト=8ビット：半角は1バイト、全角は2バイトで表すことができる
KB	キロバイト	1KB=1024バイト：K（キロ）は千倍を意味する
MB	メガバイト	1MB=1024KB=1048576B：M(メガ)は百万倍を意味する
GB	ギガバイト	1GB=1024MB=1048576KB：G(ギガ)は10億倍を意味する
TB	テラバイト	1TB=1024GB=1048576MB：T(テラ)は1兆倍を意味する

メモリを支配するCPUの正体

メモリとCPUは密接な関係にあります。メモリを理解するには、メモリを直接に支配している「CPU」を理解しておく必要があります。

「CPU」とは何か

CPU (Central Processing Unit) とは「中央演算装置」と和訳され、コンピュータの頭脳部分の役割を果たしています。特にパソコンには集積LSIを使った超小型のCPUが使われているため、MPU (Micro Processing Unit) と呼ぶ場合もあります。CPUもMPUも同義語であると考えてもよいでしょう。CPUとは大きく分けると、次の3つの仕事をするための部品です。

CPUの３つの基本機能

プログラムの実行制御

メモリ (ROM/RAM) の所定位置からプログラムを読み込んで実行します。どのプログラムのどの部分を、どのタイミングで実行するかを制御します。

演　算

実行するプログラムの命令に従って、数値演算（＋－×÷など）や論理演算（条件比較）を行います。

データの転送制御

CPU は処理を実行する際に、演算するデータを一時的に CPU 内にあるレジスタ（一時的なデータの保管場所）に保管します。CPU はこのレジスタとメモリとの間の、データ転送を制御します。また、ハードディスクやイメージスキャナなどの周辺機器とのデータ転送を制御します。

現在、パソコン用 CPU ではインテル社製（MS-DOS や Windows が動くマシンで使う）とモトローラ社製（Macintosh などで使う）の 2 系統が主流になっています。98 シリーズではインテル系 CPU が使われています。インテル系 CPU では i8086→i80286→i80386→i80486→Pentium と型番が増えるにしたがって、高速・高性能になります。また、同じ型番でもクロック周波数の大きい方が動作は高速になります。たとえば i80486（33MHz）と i80486（66MHz）では、後者の方が 2 倍も高速に動くことになります。クロック周波数とはパソコン各部の処理の足並みを揃えるための「手拍子」を意味します。兵隊の行進のときに「手拍子」の数を増やせば行進スピードも速くなるのと同じ理屈で、クロック周波数の数が多いほど高速処理ができます。

Column

●CISC 型 CPU と RISC 型 CPU

i80386 までは、内部の実行ユニットは典型的な CISC 型 CPU でした。CISC とは多くの命令を CPU 内の ROM に記録しておき、この命令用辞書を参照しながら処理を行うタイプの CPU です。複雑な命令を簡単に実行できる反面、処理が遅くなるという欠点があります。一方、i80486 や Pentium では RISC 型（あるいは RISC 的な）CPU として、内部の実行ユニットが大幅に変更されています。RISC は単純な命令（縮小命令セット）の専用回路を用意し、CPU そのものの構造を単純化しています。OS やコンパイラなどのソフトウェア側の負担は増えますが、CISC 型 CPU に比べて極めて高速に動作できる仕組みになっています。今後は Pentium や PowerPC など RISC 型 CPU が、CPU の主役となると思われます。

MS-DOSの宿命「1MBの壁」の謎

MS-DOSそのものが管理できるメモリ領域は1MBまでという制限があります。これはなぜでしょうか。本書の冒頭で述べたように、MS-DOSは米国IBM社がパソコン市場に新規参入するための切り札マシン「IBM-PC」用に開発されたOS（オペレーティングシステム=基本システム）です。このIBM-PCには、当時としては先進的かつ高速な16ビットCPU「i8086」（米国インテル社が開発）が搭載されていました。

☆i8086CPU（写真提供：インテルジャパン）

CPU（中央演算装置）には「アドレスライン」と呼ばれる端子があり、これがメモリと直結しています。CPUはこのアドレスライン1本1本に電圧をかけ、アクセスするメモリの番地（アドレス）を指定します。つまりCPUの命令を受けて荷物（データ）を運ぶ宅配便のトラックに、配送先の住所を知らせるのがアドレスラインの役割です。i8086にはアドレスラインが20本出ています。アドレスラインの端子1本が1ビットに対応するので、2^{20}=1048576バイト=1024KB=1MBとなり、すなわち1MBの範囲で番地を指定できる仕組みになります。これがMS-DOSの基本的な骨組みとなりました。互換性を維持するため、歴代のMS-DOSはこの「1MBのメモリ限界」を継承しています。

その後のインテル社のCPUはi80286、i80386、i80486、Pentiumと進化し、より広いメモリ空間をアドレスできるようになりました。i80286以上のCPUは下位のCPUと互換性を保つために、動作モードを一時的に変更できるようになっています。i8086系CPUとして動作させるモードを「リアルモード」、本来のCPUの能力を発揮させる動作モードを「プロテクトモード」と言います。

CPU	アドレスライン	アクセス可能なメモリ範囲
i8086 系	20 本	1MB
i80286 系	24 本	16MB
i80386/486 系	32 本	4096MB
Pentium	32 本	4096MB

※ここでは実際のアドレスラインとバイトイネーブル（特殊な信号線）を組み合わせた、アドレスラインの本数を表示しています

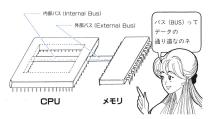
このとき i80286 以上の CPU を搭載したパソコンを使えば、メモリ問題はすべて解決しそうに見えます。ところが、MS-DOS は i8086 系 CPU（16 ビット CPU）専用の OS としての基本設計を貫いているため、32 ビット CPU を持つパソコンを使っても、起動時に強制的にリアルモード（i8086 系 CPU として）で動作させます。したがって、アドレスラインの 21 本目以降は未使用になってしまい、相変わらず 1MB のメモリ限界が存在するのです。

アドレスラインの役割

この解決には、MS-DOS 動作中でも必要に応じて瞬間的にプロテクトモードに切り替えて、プロテクトメモリを使えるように工夫します。それを行うのが「メモリマネージャ」というプログラムです。メモリマネージャには MS-DOS に付属する HIMEM.SYS/EMM386.EXE のほかに、MELWARE（メルコ）、MemoryServer（アイ・オー・データ機器）などがあります。一般的にはメモリマネージャはデバイスドライバとして提供され、CONFIG.SYS に登録します。CONFIG.SYS の設定次第でメモリ環境が左右されるのはこのためです。

32ビットCPUとは

CPUの性能を区別するときに「32ビットCPU」「16ビットCPU」などという単語をよく使います。この「○○ビットCPU」とは何を意味するのでしょうか。それはCPUの内部構造を知ることで理解できます。

☆内部バス幅と外部バス幅

CPUはプログラムを実行したり演算を実行する際に、メモリとの間でデータのやりとりを行います。このときデータが通る道を外部データバス(External Bus)と言います。また、CPU内部に取り込んだデータを処理するための、データの通り道を内部データバス (Internal Bus) と言います。

何ビットCPUであるかは、このデータバスの幅で判断します。データバスは幅が太いほど、同じ時間で一度に運べる量は多くなります (すなわち、処理速度が高速になる)。たとえば同じ量の荷物を輸送するときに、1トントラックよりも2トントラックの方が、一度に運べる量が2倍になるのと同じです。

一般的には内部データバスが32ビット幅であれば、「32ビットCPU」と呼びます (ただし正確な定義は存在しない)。内部データバスと外部データバスの双方が32ビット幅であれば、文句なしの32ビットCPUになります。

☆CPU とバスの幅

CPU	80286	80386SL	80386SX	80386DX	80486SX	80486DX	Pentium
内部バス	16	32	32	32	32	32	32
外部バス	16	16	16	32	32	32	64
アドレスバス	24	25	24	32	32	32	32

※数値単位はビット

☆Pentium（写真提供：インテルジャパン）

インテル系CPUの最新版である「Pentium」では外部データバスには64ビット幅、内部データバスには32ビット幅を採用しています。すなわち64ビットCPUに極めて近い32ビットCPUであると言えます。またPentiumには310万個のトランジスタが集積され、100MIPS（MIPSは処理速度の単位で、1MIPSは1秒間に100万回の命令を実行できる）以上の処理能力を持っています。i80486が120万5千個のトランジスタ、40MIPS（50MHz版の場合）の処理能力であるのと比べても、Pentiumが格段に高速・高機能なCPUであることがよくわかります。

しかし、現在のパソコンは基本的なアーキテクチャ（設計思想）が16ビットマシンの域を出ていないため、せっかくの高速CPUの能力を100％発揮させることがむずかしいのが現状です。実際にPentium搭載マシンとi80486搭載マシンとを比べても体感的なスピード差はそれほどありません。今後はパソコン・メモリ・周辺機器が、CPU能力に見合うだけの高速化・高機能化をどう進めるかという点が注目されます。またそれを生かすためにはOS（MS-DOSやWindows）の進化が必要であることは、言うまでもありません。

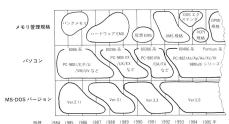
メモリ管理規格の種類を知る

MS-DOSは「1MBの壁」を超えるために、さまざまな工夫を生み出してきました。ここではそれらのメモリ管理規格の種類と仕組みについて解説します。

1MBの壁に挑んだメモリ管理の歴史

MS-DOSシステムやアプリケーションはバージョンアップを重ねるたびに、そのプログラムは肥大化しています。ところがMS-DOSが管理できる1MBのメモリ空間は、MS-DOS誕生以来変わっていません。高機能なアプリケーションを動かすには、どうしても1MBを超えるメモリを使用する必要が出てきました。

そこでEMS規格やXMS規格など、さまざまな手法が考えられてきました。8086系マシンではCPUに依存しないハードウェアEMS方式が主流でした。現在では32ビットマシンの普及にともない、プロテクトメモリをダイナミックに使うVCPI/DMPI規格がメモリ活用の主流となりつつあります。

メモリ管理規格の変遷

古典的な手法「バンクメモリ方式」

i8086系CPUを積んだ16ビットマシンでは、最初に登場したメモリ管理規格が「バンクメモリ方式」です。バンクメモリ方式は、当時は各社がバラバラな規格で行っていましたが、アイ・オー・データ機器によって提唱された「I・Oバンク方式」(BMSとも言う)がバンクメモリ規格の標準となりました。

I・Oバンク方式は、コンベンショナルメモリの上限128KBをバンクメモリにアクセスするための「窓」に使っています。この「窓」からバンクメモリ(これも128KBずつの区画に分かれている)を、必要に応じて切り替えながらデータにアクセスする仕組みになっています。増設したメモリを次々に128KBの枠に切り替える作業は、メモリボード自身が行っています。これによって、MS-DOSからはあたかも連続したメモリ空間に見えるようになっています。

バンクメモリ方式で確保したメモリは、主にRAMディスク/キャッシュディスク/プリンタスプーラなどのデータ領域として使用されていました。このバンクメモリ方式の概念・手法は、後のEMS規格の基礎になりました。

p.53

バンクメモリの仕組み

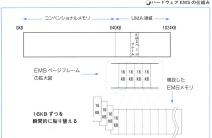
16ビットマシンに対応した「ハードウェア EMS 方式」

EMS 規格は Expanded Memory Specification の略で、ロータス/インテル/マイクロソフトの3社が提唱したメモリ管理方式です。UMA 領域の一部（通常は C0000~CFFFF に設定する）に 64KB のページフレームと呼ばれる「窓」を作り、ここから EMS メモリにアクセスします。64KB のページフレームはさらに4つに分割（16KB 単位）され、それぞれが独立して EMS メモリを貼り替えます。

基本的な原理はバンクメモリと同じで、EMS メモリボード自身がページフレームへのメモリ貼り替えを制御します。これを「ハードウェア EMS 方式」と言い、i8086 系 CPU を積んだ 16 ビットマシンのための EMS 利用方法です。したがって「EMS ボード」という専用のメモリボードを購入する必要がありました。32 ビットマシンでは CPU 自身の能力（ページング機能）を使えるようになっているので、ハードウェア EMS は過去の方式と言えるでしょう。

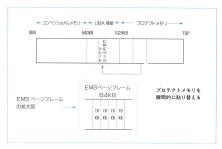
EMS メモリの用途はバンクメモリのようなデータ領域（RAM ディスク/キャッシュディスク/プリンタスプーラなど）としての使い方の他に、大きなプログラムの一部を一時的に置くための領域としての使い方があります。

参照 p.53

ハードウェア EMS の仕組み

32CPUの機能を生かした「仮想EMS方式」

「ハードウェアEMS方式」では、専用のEMSメモリボードを増設しないとEMS機能を使うことができません。ところが32ビットCPUではCPU自身が持つ能力（仮想86モード/ページング機能）を使って、プロテクトメモリをEMSページフレームに瞬間的に貼り付けることができます。

EMSページフレームにメモリを次々に貼り替えるメカニズムはハードウェアEMSと同じですが、仮想EMSの方がアクセススピードははるかに高速です。しかも、アプリケーション側からはEMS方式（ハードウェア/仮想）を意識する必要はありません。「EMS対応」のアプリケーションであれば、どのEMS方式であろうと問題なく使用できます。

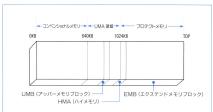
仮想EMS方式が登場した初期には、プロテクトメモリの使い方は各メモリソフトメーカーによってバラバラでした。したがって複数のデバイスドライバをCONFIG.SYSに登録した場合、設定次第ではプロテクトメモリが重複使用されてしまう危険性がありました。その後、XMS規格（MS-DOS Ver.5.0で正式採用）が発表されると、ほとんどのメモリメーカーはXMS規格に準拠した形で仮想EMSを実現するようになり、前述のようなメモリ重複使用などのトラブルはなくなりました。

DOSモードでのメモリ活用規格「XMS規格」

XMS (eXtended Memory Specification) とは、ロータス/インテル/マイクロソフト/AST の 4 社が提唱したプロテクトメモリの利用方法を定めた規格です。MS-DOS Ver.5.0A に付属している「HIMEM.SYS」というデバイスドライバは、「XMS Ver.2.0」に準拠しています。16MB 以上のプロテクトメモリを使用するにはXMS Ver.3.0以上(MELWARE for Windows/Memory ServerII/Windows3.1 で対応) を使用する必要があります。

XMS 規格に準拠したメモリドライバ (EMS ドライバ/RAM ディスク/キャッシュディスクなど) であれば、プロテクトメモリの重複使用というアクシデントを避けることができます。しかし XMS 規格は EMB をプログラム領域として使用できないなど、本来の 32 ビット CPU を生かした仕組みではありません。主に MS-DOS モードで動くアプリケーションが活用するためのメモリ規格という位置付けです。XMS 規格では次の 3 種類のメモリを活用するための、インターフェイスを定義しています。

XMS が提供するメモリ (UMB/HMA/EMB) の位置

■UMB(Upper Memory Block)

p.37

UMB とは、UMA 領域の一部にプロテクトメモリをマッピング(割り当て)して得られる RAM のことです。UMA 領域は MS-DOS システムが専用に使用する目的で作られた領域で、システム BIOS/VRAM (画面表示用のメモリ)

/ハードディスクなどの制御用のROMが割り当てられています。このうち、C0000~DFFFFまでの128KBは機器構成によっては、かなりの部分が未使用領域となっています。この未使用領域をRAMに割り当てるのがUMBです。

ただしハードディスクやEMSなどを使用していた場合は空き領域は細かく分断されているために、約48KB~64KBくらいのUMBしか確保できないのが普通です。UMBはデバイスドライバやTSR（常駐型プログラム）を専用コマンドでロードするという使い方をします。

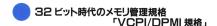
■HMA(High Memory Area)

HMAとはMS-DOSの管理領域の上限である1024KBのすぐ上の約64KBのエリア（プロテクトメモリの先頭）を指します。以前にもMEMORY-PRO386などの仮想EMS方式で、HMAを利用するメモリ設定ソフトはありました。MS-DOS Ver.5.0AではMS-DOSのシステムの一部をこのHMAに置くことができます（CONFIG.SYSに「DOS=HIGH」を指定）。HMAは使用できるプログラムが1つだけという制限があります。たとえHMAにまだ空き容量があっても、最初に使用宣言を行ったプログラムのみしか使用権が与えられません。

HMAは通常はMS-DOSシステムの格納用に使用します。64KBというと中途半端な狭い領域に感じますが、コンベンショナルメモリを専有するはずのMS-DOSのシステムの一部がここに移動するということで、その効果は絶大です。

■EMB(Extended Memory Block)

プロテクトメモリの中でHMAの上につづくメモリ領域をEMBと言います。その容量から見て、EMBはXMSの提供するメモリの主役です。XMS管理下で動くRAMディスク/EMS/キャッシュディスク/UMBなどは、EMBからメモリの提供を受けます。32ビットCPUでは最大4GBまでのEMBを認識することができます。もっともそれは論理値であって、98シリーズでは設計上の制限から、約14.6MB~127.6MBが上限です（上限は機種によって異なる）。また、32ビットCPUを積んだ98シリーズでは、標準で1MB~7.6MBのプロテクトメモリを搭載しています。

32 ビット時代のメモリ管理規格「VCPI/DPMI 規格」

プロテクトメモリを本格的に活用し、大規模でビジュアルなプログラムを動かすには、VCPI/DPMI というメモリ規格を利用します。

■VCPI(Virtual Control Program Interface)

VCPI 規格とは、32 ビット CPU をプロテクトモードで動かすときに、メモリを安全に利用するための規格です。前述の XMS 規格では EMB にはプログラムを置けないため、もっと積極的にプロテクトメモリを使いたいというニーズがありました。そのために「DOS エクステンダ」と呼ばれる製品が生まれ、プロテクトモードでの動作全般をサポートできるようになりました。DOS エクステンダを使うと、通常の MS-DOS アプリケーションを作るのと同じような工数で、手軽にプロテクトモード対応のプログラムを作れるメリットがあります。

しかしこの場合、MS-DOS の管理下にないプロテクトメモリにプログラムやデバイスドライバが常駐するため、メモリの重複使用などのトラブル回避が最大の課題になります。そこでプロテクトモードにおけるメモリ使用や従来の EMS ドライバなどとの共存をはかるために決められたのが、VCPI 規格です。VCPI 規格は仮想 8086 モードを使用するため、80386 以上の CPU が必要になります。一太郎 Ver.5 はこの VCPI 規格に対応しています。

■DPMI(DOS Protected Mode Interface)

DPMI 規格は VCPI 規格の概念をさらに広げて、マルチタスクにも対応したメモリ管理規格です。VCPI 規格はいわゆるシングルタスク専用のメモリ管理しかできません。DPMI 規格は Windows などのように、複数のアプリケーションを同時に動かすような環境のために作られました。Windows3.1 は DPMI Ver.0.9 に対応しています。

MS-DOS Ver.5.0A では HIMEM.SYS や EMM386.EXE は DPMI 規格に対応していませんが、DPMI コマンドを実行することで DPMI サーバを常駐できます。DPMI サーバを常駐すれば、以降はメモリ活用は DPMI 規格の管理下におかれます。

拡張メモリの用途を知る

XMS/VCPI/DPMIなどはプロテクトメモリをMS-DOSから使用するための規格です。そうやって使用できるようになったプロテクトメモリは、どんな用途に使えるのでしょうか。大きく分けると、次の2通りのメモリの使用法があります。

■システムが自動的に使う

一太郎Ver.5などのアプリケーションの多くは、必要なメモリを起動時に自動的に確保します。たとえばLotus1-2-3R2.4Jや松Ver.6などは起動時にEMSメモリを発見すると、勝手にEMSメモリ上にプログラム領域やデータ領域を展開します。また一太郎Ver.5やWindows3.1は起動時にプロテクトメモリに、勝手にプログラム領域やデータ領域を展開します。

32ビットマシンの普及に伴い、MS-DOSのアプリケーションは、EMSメモリを必要とするタイプからプロテクトメモリを必要とするタイプに、その製品が増えつつあります。

■ユーザーが目的をもって使う

一方、ユーザーが特定の目的のためにプロテクトメモリを割り当てることもできます。たとえば次のような用途です。

メモリ上にドライブを作る（RAM ディスク）

メモリ上に作成したドライブを「RAM ディスク」と言います。「D ドライブ」などのドライブ名が割り当てられ、通常のドライブとしてデータの保管ができます。物理的なアクセス動作を伴わないために、極めて高速にデータをやりとりできます。ただし電源を OFF（またはリセット）にすると、内容は消えてしまいます。この特徴を生かして、RAM ディスクはテンポラリデータ（アプリケーションやコマンドの一時的な作業領域）や辞書ファイルなどの、高速なアクセスが要求されるデータの格納に使われます。

主な RAM ディスクドライバには次のものがあります。

ファイル名	ドライバが含まれる製品名
RAMDISK.SYS	MS-DOS Ver.5.0/Ver.5.0A
RAMDRV.SYS	Windows3.1
EMSDISK.SYS	一太郎 Ver.5/花子 Ver.3/三四郎/五郎
EXDISK.EXE	MELWARE for Windows
IOS10.EXE	MemoryServer II

ディスクアクセスを高速にする（ディスクキャッシュ）

ハードディスクやフロッピーディスクは、メモリに比べてアクセススピードは格段に遅くなります。少しでも高速にアクセスするために設定するのがディスクキャッシュです。ディスクキャッシュはメモリ上に作成された「データの一時的な貯蔵タンク」です。

ディスクのデータを読み込む際に、このディスクキャッシュに一時的にそのデータを溜めておきます。そして次回ディスクをアクセスする際に同じデータがディスクキャッシュにあれば、ディスクをアクセスせずに、キャッシュ内のデータを CPU に返します。これによってハードディスクへのアクセス回数が減り、結果的に高速なディスクアクセスが実現します。

主なディスクキャッシュドライバには、次の種類があります。

ファイル名	ドライバが含まれる製品名
SMARTDRV.SYS	MS-DOS Ver.5.0/Ver.5.0A
SMARTDRV.EXE	Windows3.1
EMSCACHE.SYS	一太郎 Ver.5/花子 Ver.3/三四郎/五郎
HYPERDSK.EXE	MELWARE for Windows
DC10.EXE	MemoryServer II

プリントアウト作業を高速にする（プリンタスプーラ）

プリンタへの出力はもっとも時間がかかる作業です。通常、アプリケーションは印刷が完了するまで、他の作業ができなくなります。そこで印刷データを一時的に溜めておき、アプリケーションで次の作業をすぐに行えるようにするのがプリンタスプーラです。プリンタスプーラには複数の印刷データを溜めておくことができ、プリンタの作業状況に応じて印刷データをプリンタに送ります。

ただし最近のページプリンタなどは、プリンタ本体が持つバッファ容量が大きく、1~2 ページ分の印刷データならプリンタスプーラがなくても問題がありません。また、アプリケーションとの相性が問題になるケースがあり、現在ではプリンタスプーラはあまり使われていません。主なプリンタスプーラには、MELWARE Ver.5 の MELPRIN.COM があります。

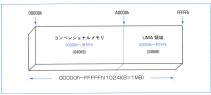
メモリの状態を調べる

メモリがどのような使われ方をしているかを調べる方法について説明します。

アドレスの読み方

CPUがメモリにアクセスするときには、アドレス（番地）をたよりに行います。ちょうど家の1軒ずつに番地が付けられていると同じで、メモリも1バイト単位でアドレスが定義してあります。ただしコンピュータ内部はすべてのデータをデジタル（2進数）で扱うので、アドレスも2進数になります。しかし、0と1だけの2進数で表したアドレスは、人間が簡単に読み取れる形ではありません。そこで人間が理解しやすいように、16進数でアドレスを表記するのが一般的になっています。

たとえば「C0000hからCFFFFhまでにEMSページフレームを設定」などというように記述したときの、「C0000h」や「CFFFFh」がメモリのアドレスです。「h」はHexadecimal（16進数）の頭文字で、このアドレスが16進数表記であることを表します（「h」を省略することもある）。

☆アドレスを16進数で表記する

16進数は0・1・2・3・4・5・6・7・8・9・A・B・C・D・E・Fの16個の文字で表します。Fを超えると次に繰りあがります。たとえば「F」の次は「10」

になります。この方法で表すと、コンベンショナルメモリは00000h~9FFFFh、UMA 領域はA0000h~FFFFFhになります。そしてプロテクトメモリは100000hから開始されます。

10進数	2進数	16進数	10進数	2進数	16進数	10進数	2進数	16進数
0	0	0	10	1010	A	20	10100	14
1	1	1	11	1011	B	21	10101	15
2	10	2	12	1100	C	22	10110	16
3	11	3	13	1101	D	23	10111	17
4	100	4	14	1110	E	24	11000	18
5	101	5	15	1111	F	25	11001	19
6	110	6	16	10000	10	26	11010	1A
7	111	7	17	10001	11	27	11011	1B
8	1000	8	18	10010	12	28	11100	1C
9	1001	9	19	10011	13	29	11101	1D

なお、パソコン雑誌などで「EMSページフレームはC000から始まる」などというようにメモリアドレスを4桁で表記しているのを見ることがあります。これは下1桁を省略した表記方法で、「C0000h」と同じことを表します。正確にはセグメンテーションという i8086系CPU独自の論理アドレスの指定法が元になっていますが、要するに末尾に1桁目と同じ数字を1個付ければ正式な物理アドレスになります。セグメンテーションはセグメントアドレス(上位4桁)とオフセットアドレス(下位4桁)を使って、5桁のアドレスを表すという変則的な方法を用います。これはi8086が誕生したときに8ビットCPUである i8080との互換性を保つためのものがまだ残っているのです。「C000」などはこのセグメントアドレスだけを簡略的に表記している訳です。

Column

●16進数を10進数に換算する

16進数で表記されたメモリアドレスを10進数に換算するには、関数電卓/表計算ソフトの関数(三四郎・Lotus1-2-3ではDECIMAL関数、EXCELではHEX2DEC関数)やSYMDEBコマンド(MS-DOS拡張機能セットに含まれる)を使う方法があります。また計算式から算出することもできます。

たとえば「7FF3h」を10進数にするには、$16^3 \times 7 + 16^2 \times 15 + 16^1 \times 15 + 3$ を計算します。すなわちそれぞれの桁に16の「桁数-1」乗の値を掛け算し、桁ごとに算出した値を足し算すればよいわけです。

メモリの使われ方を調べる

「メモリが足りません」などというメッセージが表示されたときに、一体何が原因で足りなくなっているのかを調べるには「メモリマップ」を表示します。また UMB を有効に使うには、やはりメモリマップで目的のプログラム・デバイスドライバの必要な容量を調べる必要があります。メモリマップとはメモリが現在どのように使われているかを表示したものです。CONFIG.SYS を極めるためにはメモリの使われ方を調べるテクニックが必須になります。

メモリマップを表示するための代表的な 3 つのツールを例に、その使い方と読み方を説明します。

■VMAP.COM

VMAP.COM は Vz エディタ（販売：ビレッジセンター、作者：兵藤嘉彦氏）に付属するメモリ表示ユーティリティです。またパソコン通信（NIFTY-Serve など）で無料入手できるフリーソフトでもあります。VMAP.COM を起動するには、次のように入力します。

```
A〉VMAP⏎
```

※「VMAP /?⏎」と入力すると、ヘルプ画面を表示できます

これで次ページのようなメモリマップが表示されます。UMB メモリ、コンベンショナルメモリ、EMS メモリ、XMS メモリの使用状況をそれぞれ調べることができます。「addr」は該当プログラムが使用するメモリの先頭のアドレス（セグメントアドレス）を表示します。「size」は該当プログラムの使用しているメモリ容量をバイト（Byte）単位で表示します。「owner/parameters」は該当するプログラム名が表示されます。ここに〈free〉と表示される場合は、その部分が空き領域になっていることを表します。

たとえば、UMB に常駐している「diskxii0」は 49072 バイト（約 49KB）を消費しているのがわかります。またコンベンショナルメモリの空き容量は 535232 バイト（約 535KB）であることがわかります。

コンベンショナルメモリ　UMA 領域　プロテクトメモリ

0KB　640KB　1024KB　1408KB

HSB
VMM386
DCIO
DISKX
XDRV
ATOK8 など

UMB

HMA

アドレス　サイズ　プログラム

```
VMAP Version 2.01  Copyright (C) 1989-91 by c.mos

addr PSP  blks   size  owner/parameters                 hooked vectors
---- ---- ---- ------  ---------------------------      ---------------------------
D003 <--     1  49072  diskxii0                          0B
DD03 sys     1   4128  kkcfunc
DE06 sys     1   3488  atok8ex
DEE1-E000    1   4576  <free>
                       --- UMB total:  60 KB ---
0586 sys     1   2688  hsb                               09
062F sys     1   3376  vmm386                            1F 4B 67
0703 sys     1   9760  dc10                              13 1B
0966 sys     1   4944  diskx                             20 26
0A9C sys     1    464  xdrv
0ABA sys     1    352  atok8a
0AD1 sys     1  15680  atok8b CON                        18 29 6F
0EA6 sys     4  25024  <config>
14C6 <--     4   3024  command                           22 23 24 2E
1587 <--     1  14048  share /L:500
18F6         1    288  <free>
1909 <--     1   6320  doskey Z=ZIMCUT
1A95 <--     1  11216  mirror /tA /tB /tC                19 21 2F
1D53-9FFF    1 535232  <free>

----- EMS ver4.0 (frame: C000h) -----             ----- XMS ver3.00 -----
handle pages   size  name                         HMA used:  59 KB by DOS
------ -----  ------ --------                     EMB free: 10784 KB
     1   192  3072k  XMS
     2     0     0k  VCPI
     3     4    64k  ATOK8A
     4     2    32k  ATOK8B
     5     3    48k  ATOK8C
     6     3    48k  ATOK8D
  free   674 10784k
 total   878 14048k
```

UMB を使用するプログラム一覧

コンベンショナルメモリを使用するプログラム一覧

HMA を使用するプログラム

EMS メモリを使用するプログラム一覧

■MEM.EXE

MEM.EXEはMS-DOS Ver.5.0以上に付属するメモリ表示ユーティリティです。MEM.EXEを起動するには次のように入力します。

A〉MEM /P ¦ MORE ↵

※「MEM /? ↵」と入力すると、ヘルプ画面を表示できます

これで次ページのようなメモリマップが表示されます。コンベンショナルメモリ、UMBの使用状況をそれぞれ調べることができます。MEMコマンドはメモリ内部の詳細な情報を表示でき、UMBの内容を詳しく調べるときなどには便利です。これらの情報は複数画面に渡って表示されるので、上記の入力例ではMS-DOSのパイプ機能（¦記号）を使って、1画面ごとに停止するようにしてあります。

MEMコマンドの表示項目には、次のような意味があります。

アドレス

「名前」に表示してあるプログラムが使用するメモリの開始アドレスが、16進数で表示されます。MEMコマンドの表示ではコンベンショナルメモリとUMBの境がわかりづらいという欠点があります。A0000h（UMBの開始位置）を目安に、それより上位（次ページでは0D0010より下に表示される部分）がUMBで、それより下位（次ページでは09FFF0より上に表示される部分）がコンベンショナルメモリです。

名　前

メモリを使用しているプログラム名を表示します。MS-DOS自身が確保した領域には名前が表示されないものが希にあります。

サイズ

p.56

プログラムが使用するメモリ容量を16進数で表示します。

タイプ

メモリを使用するプログラムの種別を表示します。「インタラプトベクタ」「ROMコミュニケーションエリア」「システムデータ」「システムプログラム」は、MS-DOSシステムまたはハードウェアが使用します。「DEVICE=」はCONFIG.SYSで組み込んだデバイスドライバです。

アドレス	名前	サイズ	タイプ
000000		000400	ｲﾝﾀﾗﾌﾟﾄ ﾍﾞｸﾀ
000400		000200	ROM ｺﾐｭﾆｹｰｼｮﾝ ｴﾘｱ
000600	IO	003D60	ｼｽﾃﾑ ﾃﾞｰﾀ
004360	MSDOS	0014E0	ｼｽﾃﾑ ﾃﾞｰﾀ
005840	IO	00F400	ｼｽﾃﾑ ﾃﾞｰﾀ
	HSB	000A80	DEVICE=
	VMM386	000D30	DEVICE=
	DC10	002620	DEVICE=
	DISKX	001350	DEVICE=
	XDRV	0001D0	DEVICE=
	ATOK8A	000160	DEVICE=
	ATOK8B	003D40	DEVICE=
		0006F0	FILES=
		000050	FCBS=
		005190	BUFFERS=
		0008F0	LASTDRIVE=
014C50	COMMAND	000A50	ﾌﾟﾛｸﾞﾗﾑ
0156B0	MSDOS	000040	-- 空き --
015700	COMMAND	000100	環境
015810	MSDOS	000040	-- 空き --
015860	SHARE	0036E0	ﾌﾟﾛｸﾞﾗﾑ
018F50	MEM	000110	環境
019070	MSDOS	000000	-- 空き --
019080	DOSKEY	0018B0	ﾌﾟﾛｸﾞﾗﾑ
01A940	MIRROR	002BD0	ﾌﾟﾛｸﾞﾗﾑ
01D520	MEM	017640	ﾌﾟﾛｸﾞﾗﾑ
034B70	MSDOS	06B470	-- 空き --
09FFF0	ｼｽﾃﾑ	030010	ｼｽﾃﾑ ﾌﾟﾛｸﾞﾗﾑ
0D0010	IO	00BFC0	ｼｽﾃﾑ ﾃﾞｰﾀ
		00BFB0	
0DBFE0	ｼｽﾃﾑ	001020	ｼｽﾃﾑ ﾌﾟﾛｸﾞﾗﾑ
0DD010	IO	001DE0	ｼｽﾃﾑ ﾃﾞｰﾀ
	KKCFUNC	001020	DEVICE=
	ATOK8EX	000DA0	DEVICE=
0DEE00	MSDOS	000120	-- 空き --
0DEF30	MSDOS	0010C0	-- 空き --

■MSD.EXE

MSD.EXE は Windows3.1 に付属するパソコン環境の表示ユーティリティです。本来は、マイクロソフト社のサポート要員が、パソコン環境を調査するときに使用するために作られたものです。メモリ情報だけでなく、CPU や周辺機器などの総合的な情報を調べることができる点が特徴です。たとえばコンベンショナルメモリと UMB の内容を調べるには、次のように操作します。

操 作

①MSD コマンドの起動

MS-DOS のコマンドラインから「MSD⏎」と入力します（MSD メニューの表示）。

※「MSD /?⏎」と入力すると、ヘルプ画面を表示できます

②［T：TSR の状態］の選択

・T と入力します。

・メモリマップが表示されるので、↓↑キーでスクロールさせます。

③MSD コマンドの終了

f·3 キーを押します。

☆MSD メニュー

TSRの状態

プログラム名	ｱﾄﾞﾚｽ	ｻｲｽﾞ	コマンド ライン パラメータ
システム データ	0584	62464	
HSB	0586	2688	HSBDRV12
VMM386	062F	3376	EMMXXXX0
DC10	0703	9760	$DC10$E$
DISKX	0966	4944	ﾌﾞﾛｯｸ ﾃﾞﾊﾞｲｽ
XDRV	0A9C	464	XDRV2000
ATOK8A	0ABA	352	ATOK8A%%
ATOK8B	0AD1	15680	CON
ﾌｧｲﾙ ﾊﾝﾄﾞﾙ	0EA6	1776	
FCBS	0F16	80	
BUFFERS	0F1C	20880	
ディレクトリ	1436	2288	
COMMAND.COM	14C5	2640	/F C:3.TXT
空きメモリ	156B	64	
COMMAND.COM	1570	256	/F C:3.TXT
空きメモリ	1581	64	
M	1586	14048	/L:500
DK6.EXE	18F5	240	
DK6.EXE	1905	12432	
MSD.EXE	1C0F	288	/F C:3.TXT
MSD.EXE	1C22	6320	Z=ZIMCUT
MIRROR.COM	1DAE	11216	/tA /tB /tC
MSD.EXE	206C	330448	/F C:3.TXT
MSD.EXE	711A	8192	/F C:3.TXT
空きメモリ	731B	183856	
除外UMB領域	9FFF	196624	
システム データ	D001	49088	
除外UMB領域	DBFE	4128	
システム データ	DD01	7648	
KKCFUNC	DD03	4128	KKCFUNC$
ATOK8EX	DE06	3488	MS$KANJI
空きメモリ	DEE0	288	
空きメモリ	DEF3	4288	

実際にメモリを増設する

ここではメモリを増設する際に覚えておくべきポイントを説明します。

どれくらいのメモリを増設するか

32ビットマシンで「メモリを増設する」と言うと、プロテクトメモリ（エクステンドメモリ）を増やすことを意味します。16ビットマシンでは「EMS用メモリボード」「バンクメモリボード」などのように、用途別のメモリボードを使用するので、混同しないように注意が必要です。32ビットマシンで実現する仮想EMS・仮想UMB・仮想バンクメモリ・RAMディスク・ディスクキャッシュなどは、すべてプロテクトメモリからメモリ供給を受けます。またWindowsや一太郎Ver.5は、プロテクトメモリ上に直接にプログラムを置いて動作します。このように32ビットマシンにはプロテクトメモリは必須です。

使用しているパソコンにどれだけのプロテクトメモリが搭載されているかは、パソコンの電源をON（またはリセットボタンを押す）にしたときに画面左上に表示される数字で確認できます。次の表示例では、8192KB（約8MB）のプロテクトメモリがパソコンに搭載されていることがわかります。

では実際にどれくらいのプロテクトメモリを用意すれば、パソコンが快適に動くのでしょうか。MS-DOS対応アプリケーション（一太郎Ver.4.3/Lotus1-2-3R2.4J/桐Ver.5/松Ver.6など）だけを使うなら、プロテクトメモリは1~3MBで十分です。この場合はパソコンに標準で実装されているプロテクトメモリ（次ページの表参照）だけで間に合うでしょう。

しかしWindows3.1以上を使うには最低でも4MB以上、快適に使うには8

機種名	標準搭載メモリ	増設上限メモリ
PC-9821Bs2/U2、M2、U7	3.6MB	19.6MB
PC-9801BX2/U2、M2	1.6MB	17.6MB
PC-9801BX2/U7	3.6MB	19.6MB
PC-9821Be/U7W	5.6MB	35.6MB
PC-9821Ba2/U7	3.6MB	19.6MB
PC-9821Bf/U8W	7.6MB	71.6MB
PC-9821Es	5.6MB	37.6MB
PC-9821Xa	7.6MB	127.6MB
PC-9821Xt	15.6MB	255.6MB
EPSON		
PC-486GF シリーズ	1.6MB	14.6MB
PC-486GR シリーズ	1.6MB	14.6MB
PC-486P2 シリーズ	1.6MB	14.6MB
PC-486GRS シリーズ	3.6MB	64MB
PC-486GRP シリーズ	3.6MB	64MB
PC-486PWIN	3.6MB	14.6MB
PC-486MR2 シリーズ	3.6MB	61.6MB
486MU2 シリーズ	5.6MB	61.6MB

☆主なノート型パソコンの標準搭載メモリ

機種名	標準搭載メモリ	増設上限メモリ
NEC		
PC-9801NS シリーズ	1.6MB	11.6MB
PC-9801NS/R シリーズ	1.6MB	14.6MB
PC-9801NA シリーズ	3.6MB	14.6MB
PC-9801NX/C	1.6MB	13.6MB
PC-9801NX/C120	3.6MB	14.6MB
PC-9801NL	3.6MB	11.6MB
PC-9801NS/A	3.6MB	19.6MB
PC-9821Ne	3.6MB	14.6MB
PC-9821Ne120/W	5.6MB	14.6MB
PC-9821Np	5.6MB	37.6MB
PC-9821Ns	5.6MB	37.6MB
PC-9821Nd	5.6MB	37.6MB
EPSON		
PC-386NOTEAR シリーズ	1.6MB	9.6MB
PC-386NOTEARC シリーズ	1.6MB	9.6MB
PC-386NOTEARX シリーズ	1.6MB	9.6MB
PC-386NOTEAS シリーズ	1.6MB	17.6MB
486NAU シリーズ	3.6MB	35.6MB
486NASD2	5.6MB	33.6MB

操　作

①内部増設メモリの用意

購入した内部増設型メモリボードに増設 RAM サブボード（メーカーによってはSIM モジュールと呼ぶ場合がある）を装着します。

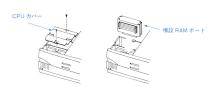
②本体カバーの取り外し

・パソコンの電源を OFF にして、電源ケーブルを抜きます。
・本体カバーの背面のネジ（2 本）と側面のネジ（左右 1 本ずつ）を外します。
・本体カバーを後方に引き出してから、上に持ち上げて取り外します。

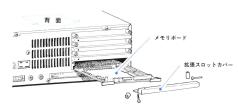
98MATE の本体カバーを取り外す

③メモリボードの取り付け

・CPU カバーのネジ（2 本）を外し、CPU カバーを取り外します。
・増設メモリボードをメモリ専用スロットに差し込みます（最後にカチッと音がするまで静かに押す）。
・CPU カバーと本体カバーを取り付けると、メモリ増設は完了します。

メモリボードを取り付ける

■拡張スロット型メモリボードの装着

「拡張スロット型メモリボード」とはパソコン背面の拡張スロットに差し込むタイプのメモリボードです。拡張スロットは初期の98シリーズの各種ボードと互換性を持たせるためのもので、CPUとは16ビット幅の伝送路で結ばれています。このため、内部増設型メモリボードに比べて半分以下の低速アクセスになります。ただし、拡張スロットはあらゆる98シリーズで使用できるため、長期的に使用できるという「安心感」があります。拡張スロット型メモリボードの装着方法は、すべての98シリーズでまったく同じです。たとえばPC-9821Asでは次の手順で装着します。

拡張スロットにメモリボードを差し込む

操 作

①拡張スロットカバーの取り外し

・パソコンの電源をOFFにして、電源ケーブルを抜きます。
・本体背面の拡張スロットカバーのネジ(2本)を外します。
・拡張スロットカバーを取り外します。

②メモリボードの装着

・マニュアルに従って、ボードのディップスイッチを設定します。
・増設メモリボードを拡張スロットに差し込みます(最後にカチッと音がするまで静かに押す)。
・拡張スロットカバーを取り付けると、メモリ増設は完了します。

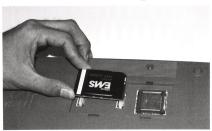
ノート型パソコンのメモリを増設する

ノート型パソコンには、「内部増設型メモリボード」「メモリカード」の2種類のメモリを装着できます。結論から言うと、32ビットパソコンには「内部増設型メモリボード」を選んだ方が、Windowsなどの動作は高速になります。

■内部増設型メモリボードの装着

「内部増設型メモリボード」とはノート型パソコン本体の下面（キーボード面の裏側）のカバーを開けて、メモリ専用のスロットに差し込むタイプのメモリボードです。デスクトップ型と同様に、CPUと32ビット幅の伝送路で結ばれているため、メモリカード（16ビット幅の伝送路）に比べて2倍以上高速にメモリアクセスできます。しかし、ノート型パソコン機種ごとに規格が異なるので、他のパソコンには流用できないという欠点があります。

たとえばPC-9801SX/Tに内部増設型メモリボードを装着するには、次のように操作します。

☆内部増設型メモリをセットする

操 作

①パソコンのセットアップ

・ノート型パソコンの電源をOFFにし、ACアダプタを抜きます。
・背面のバックアップスイッチをOFFにします。（メモリ装着後にONに戻す）

②メモリボードの装着

・ノート型パソコンを裏返しにします。
・マイナスドライバで中央にあるカバーのツメ(2カ所)を開けます。
・カバーを取り外します。
・内部増設型メモリボードを差し込みます。
・カバーを取り付けると、メモリ増設は完了します。

■メモリカードの装着

メモリカードとは、ノート型パソコン本体の左側面の汎用スロットに差し込むタイプのメモリカード(クレジットカードサイズ)です。汎用スロットはCPUと16ビット幅の伝送路で結ばれているため、内部増設用メモリボードに比べて半分以下の低速アクセスになります。ただし、メモリカードはすべてのノート型パソコン(98シリーズ)で使用できるという利点があります。またメモリカードの脱着はイジェクトボタンを押すだけで簡単に操作できます。

☆メモリカードをスロットに装着する

解　説

■増設したメモリを認識させる

次の機種に該当するノート型パソコンをお持ちの場合は、それぞれの手順で増設したメモリをパソコンに認識させてください。

NEC 98note シリーズ（PC-9801N、NS、NX/C、PC-9821Ne を除く）

①HELP キーを押しながら、電源を ON にします（システムメニューの表示）。
②［動作環境の設定］➜［その他の設定］を選択します。
③［EMS 機能］には［使用する］を選択します。
④システムメニューを終了し、MS-DOS を起動します。

EPSON PC-486NOTE AS

①HELP キーを押しながら、電源を ON にします（システムメニューの表示）。
②［拡張メモリ・記憶装置の設定］を選択します。
③入力ウィンドウが表示されるので、プロテクトメモリの総容量を入力します。
④システムメニューを終了し、MS-DOS を起動します。

EPSON PC-386NOTE AR

①HELP キーを押しながら、電源を ON にします（システムメニューの表示）。
②［RAM ドライブの設定］を選択します。
③［内部拡張メモリの使い方］には［プロテクトメモリ］を選択します。
④システムメニューを終了し、MS-DOS を起動します。

EPSON PC-386NOTE A、AE、W、WR

①HELP キーを押しながら、電源を ON にします（システムメニューの表示）。
②［環境設定］を選択します。
③［RAM ボードの使い方］には［EMS/プロテクトモードのメモリ］を選択します。
④システムメニューを終了し、MS-DOS を起動します。

徹底活用編

ここでは
MS-DOS／MemoryServer Ⅱ／MELWARE for Windows の
それぞれのメモリマネージャを使って、
UMBの有効活用や高速アクセスなどの環境を構築する方法を
徹底的に解説します

第1章 MS-DOS Ver.5.0A を徹底活用する

MS-DOS を使って最適な動作環境を設定するまでを、徹底的に解説します
ここでは、MS-DOS Ver.5.0A を前提に操作・設定を説明します

メモリを有効活用する

MS-DOS Ver.5.0Aに付属するINSTDOSコマンドを使って、MS-DOSをインストールすると、以下のようなCONFIG.SYSとAUTOEXEC.BATが自動作成されます。ここではこの2つのファイルを土台として、メモリを無駄使いしないための設定法を解説します。

☆MS-DOSインストール時に自動作成されるCONFIG.SYS

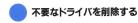
```
FILES=30
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB /T=A:¥DOS¥EXTDSWAP.SYS
DEVICEHIGH=A:¥DOS¥PRINT.SYS /U
DEVICEHIGH=A:¥DOS¥SMARTDRV.SYS 2048 128
DEVICEHIGH=A:¥DOS¥RSDRV.SYS
DEVICEHIGH=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH,UMB
```

☆MS-DOSインストール時に自動作成されるAUTOEXEC.BAT

```
@ECHO OFF
PATH A:¥DOS;A:¥
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
MOUSE
DOSSHELL
MOUSE /R
```

☆この時点でのフリーエリア

コンベンショナルメモリ	:558,656バイト
UMB総容量	:28,960バイト

不要なドライバを削除する

まず不要なデバイスドライバを CONFIG.SYS から削除しましょう。

操　作

CONFIG.SYS を次のように修正します。

修正前の CONFIG.SYS

```
FILES=30
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB /T=A:¥DOS¥EXTDSWAP.SYS
DEVICEHIGH=A:¥DOS¥PRINT.SYS /U      ←―― 削除
DEVICEHIGH=A:¥DOS¥SMARTDRV.SYS 2048 128
DEVICEHIGH=A:¥DOS¥RSDRV.SYS         ←―― 削除
DEVICEHIGH=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH, UMB
```

修正後の CONFIG.SYS

```
FILES=30
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB /T=A:¥DOS¥EXTDSWAP.SYS
DEVICEHIGH=A:¥DOS¥SMARTDRV.SYS 2048 128
DEVICEHIGH=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH, UMB
```

解　説

■フリーエリアの比較

フリーエリア	修正前	修正後
コンベンショナルメモリ	558,656 バイト	558,656 バイト
UMB 総容量	28,960 バイト	36,704 バイト

■PRINT.SYSとRSDRV.SYSの役割

p.213-217

PRINT.SYS は PC-PR201 系プリンタを制御するデバイスドライバです。RSDRV.SYS は RS-232C（通信用インターフェイス）を制御するデバイスドライバです。一部の MS-DOS コマンドではこれらのデバイスドライバを使ってデータを入出力します。しかし、ほとんどの市販アプリケーション（まいと～く・一太郎・Lotus1-2-3 など）では、独自にプリンタ制御機能や RS-232C 制御機能を持っているので、特にこれらのデバイスドライバを必要としないのが実状です。

これらのデバイスドライバを取り除くことによって、PRINT.SYS では約 5.3KB、RSDRV.SYS では約 2.4KB のメモリを節約できます。

また、NEC の「AI かな漢字変換」を使うユーザーも少ないと思われます。したがって、NECAIK1.DRV と NECAIK2.DRV を削除して目的の FEP を登録しておくとよいでしょう。

バッファ領域を小さくする

MS-DOS インストール時点では BUFFERS と FCBS の指定は省略されています。したがって省略時の既定値「BUFFERS=20」「FCBS=4」が指定されていることになります。そこでメモリを節約するために、BUFFERS=10、FCBS=1 をそれぞれ追加しましょう。

操 作

CONFIG.SYS を次のように修正します。

修正後の CONFIG.SYS

```
FILES=30
BUFFERS=10          ←――― 追加
FCBS=1              ←――― 追加
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB /T=A:¥DOS¥EXTDSWAP.SYS
DEVICEHIGH=A:¥DOS¥SMARTDRV.SYS 2048 128
DEVICEHIGH=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH,UMB
```

■フリーエリアの比較

フリーエリア	修正前	修正後
コンベンショナルメモリ	558,656 バイト	569,264 バイト
UMB 総容量	36,704 バイト	36,704 バイト

■BUFFERS 値を小さくする理由 p.185

BUFFERS はディスクバッファの容量を指定します。これは簡易ディスクキャッシュのような役割をします。ここでは SMARTDRV.SYS というディスクキャッシュで 2MB を指定しているので BUFFERS 値は 10 でも十分です。

BUFFERS で指定したディスクバッファはコンベンショナルメモリ上に作成されるので、BUFFERS の値を多く取りすぎるとフリーエリアが小さくな

ります。「BUFFERS=1」当たりのメモリ消費は、パソコンに接続されているディスク容量によって異なります。次の表を目安に考えると、ここでBUFFERS=10と指定したことで、約10KBのメモリを節約できるようになります(省略時の既定値はBUFFERS=20)。

ディスク容量	セクタ長	BUFFERS=1 当たり消費量
1MB~64MB	1024バイト	約1044バイト
65MB~128MB	2048バイト	約2068バイト
129MB~2047MB	512または256バイト	約1044バイト

■SMARTDRV.SYSのキャッシュが効かない場合

SMARTDRV.SYSは、機器環境やソフト環境によっては機能しない場合があります。

ディスクキャッシュが機能しないと、BUFFERS=10では少なすぎます(20程度が適当)。筆者の場合、ディスクキャッシュが機能しているかどうかを調べるには、Lotus1-2-3を連続して2回起動しています。2回目の起動のとき、ハードディスクのアクセスランプが付かずに高速に起動できたら、ディスクキャッシュが機能していると判断しています。

なお、Windows3.1ではSMARTDRV.SYSは機能しません。したがって、この場合はWindows3.1に付属するEMM386.EXEとSMARTDRV.EXEを使った方がよいでしょう。 p.162

■FCBS値を小さくする理由 p.191

FCBSはファイルコントロールブロック方式でファイルをオープンするときの、最大ファイル数を指定します。ファイルコントロールブロック方式はMS-DOSVer.2.11時代に使われた旧式の方式で、現在ではFCBSを使っているアプリケーションは皆無といってもよいでしょう。現在ではFILES(ファイルハンドル方式)を使うのが常識になっています。

FCBS=1当たり約58.6バイトを消費します。したがってFCBS=1(指定できる最低値)を指定したことで、コンベンショナルメモリを約176バイト節約できます(省略時の既定値はFCBS=4)。

DOSSHELL を除去する

DOSSHELL は使用しない場合は、CONFIG.SYS と AUTOEXEC.BAT から関係する指定を削除しましょう。

操 作

①CONFIG.SYS の修正

EMM386.EXE から/T オプションを取り除きます。

```
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB /T=A:¥DOS¥EXTDSWAP.SYS
                    ↓
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB
```

②AUTOEXEC.BAT の修正

AUTOEXEC.BAT から次の 3 行を削除します。

```
MOUSE
DOSSHELL
MOUSE /R
```

■/T オプションの機能

/T は MS-DOS Ver.5.0A になって追加されたオプションで、DOSSHELL のタスクスワップ時にメモリ内と作業ファイルのデータ管理を行うなど、タスクスワップを補助します。したがって、DOSSHELL を使わない環境ではこの指定は不要です。/T はメモリを消費しません。

■DOSSHELL のメモリ活用

DOSSHELL は起動時にプロテクトメモリをすべて EMB として確保してしまい、プロテクトメモリを活用するようなプログラムを同時に実行できないという欠点があります。MELWARE for Windows や MemoryServer II では DOSSHELL に分け与えるプロテクトメモリ容量を制限できますが、EMM386.EXE ではすべて DOSSHELL に取られてしまい、他のアプリケーションが起動できない場合があります。 p.263

UMBを大胆に使う

物理アドレスA0000h~FFFFFhまでの384KBのメモリ空間を「UMA領域」と言います。このUMA領域の未使用部分にRAMを割り当て、ユーザーが自由に使えるようにした領域を「UMB」と言います。ここまでは「基礎知識編」にて説明しました。p.49

ではUMBをできるだけ多く確保し、活用するにはどうしたらよいのでしょうか。

UMB活用の難しさは、「UMA/UMBの詳細な使用状況を目で簡単に確認できない」「UMBにロードすべきプログラムの必要容量があらかじめわからない」「UMBブロックが連続していないため、プログラムをロードする順番によってはUMBが使われないままになる」などという点にあります。そのため、かなり熟練したユーザーでも、何度も試行錯誤を繰り返すことになります。

ここではMS-DOS Ver.5.0Aに付属するメモリマネージャ「EMM386.EXE」を使って、UMBを有効利用するまでを、次の手順で説明します。

使用可能なUMBを調べる

現在、PC-9821As(i80486DX、33MHz)には14MBのプロテクトメモリ、IDE型ハードディスク(240MB)とSCSI型ハードディスク(270MB)が装着されています。このときのUMBの状態を調べましょう。

操　作

①UMBの総容量の表示

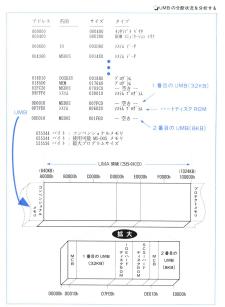
リセットボタンを押し、MS-DOSを再起動します。起動時のメッセージを見ると、使用できるUMBの総容量が39KBであることがわかります。

```
ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 5.00A-H
Copyright (C) 1981,1992 Microsoft Corp. / NEC Corporation

HIMEM: XMSドライバ Ver.2.60 - 05/10/90
XMS仕様バージョン 2.0
Copyright 1988-1990 Microsoft Corp./ NEC Corporation

64K ハイメモリ領域が有効です


EMM386ドライバ
(C) Copyright Microsoft Corp. / NEC Corporation 1986,1991

  EMSが使用可能です
    使用可能なページ数は  256 です
    使用可能なハンドル数は   64 です
    ページフレームアドレスは C000Hです
  UMBが使用可能です
    使用可能なUMBメモリは   39KB です
    使用可能な最大UMBサイズは   31KB です

Command ﾊﾞｰｼﾞｮﾝ 5.00A
A:¥>
```


②UMBの分断状況表示　 p.59

「MEM /P ¦ MORE⏎」と入力します。ここではアドレスA0000h以降(次の図では9FFF0hより下)に注目します。これを見るとUMBの総容量は39KBですが、2つのブロック(32KBと8KB)に分断されていることがわかります。

なお、UMBブロックの先頭には10h(16バイト)の「MCB(メモリコントロールブロック)」が必ずあります。MCBにはUMB位置/容量/使用状況などが記録されていて、MS-DOSはこのMCBをたよりにUMBにアクセスする仕組みになっています。メモリマップ表示ユーティリティでは、この

MCBの扱い方に違いがあります。MCBも含めてすべてUMBとしたり、MCBの表示を省略したりします。本書のメモリ概略図では、以降、MCBも含めてUMBとして表します。

UMBが確保される領域

EMM386.EXEで/UMBを指定すると、UMA領域のうちC0000h~DFFFFhまでの128KBの範囲をサーチして、空いている領域をUMBに割り当てます。通常は、C0000h~CFFFFhまでの64KBにはEMSページフレームが割り当てられます。残りのD0000h~DFFFFhまでの64KBの中にはハードディスクROMなどが割り当てられるので、通常は30KB~40KBが自由に使えるUMB容量になります。

UMA領域は何に使われているか

これまで見てきたように、UMBはUMA領域のわずか10%前後しか使うことができません。ではUMA領域には何が割り当てられているのでしょうか。正確には、パソコン機種によって千差万別ですが、一般的には次のように使われています。

デスクトップパソコンのUMA領域

※PC-9821As/U2にSCSIハードディスクを接続したケース

ノート型パソコンのUMA領域

※PC-9801NS/Tでのケース

ハードディスクROMを移動する

83ページのUMB分析では、アドレスD7FE0h~DE00Fhまでの24KBの範囲のどこかに、SCSI型ハードディスクのBIOS ROMが存在していることがわかりました。そこで隠しオプション/MOVEHDBIOSを使って、SCSI型ハードディスクのBIOS ROMをテキストVRAMの未使用領域(A6000h前後)に移動し、UMBを増やしましょう。

操 作

CONFIG.SYSを次のように修正します。

修正後のCONFIG.SYS

```
FILES=30
BUFFERS=10
FCBS=1
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB /MOVEHDBIOS /I=DC00-DFFF
DEVICE=A:¥DOS¥SMARTDRV.SYS 2048 128
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH,UMB
```

（追加するオプション → /MOVEHDBIOS）

解 説

■ハードディスクROMとは

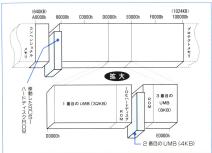
ハードディスクのBIOS ROMとは、ハードディスクを制御するためのプログラムのことです。ハードディスクのメーカーや機種によって、UMA領域のどのアドレスをBIOS ROMに使うかは千差万別です。メモリマップからその位置を推測する際には、次のアドレスを参考にするとよいでしょう。

ハードディスク種別	使用アドレス	使用容量
SASI型ハードディスク	D7000h~D7FFFh	4KB
IDE型ハードディスク	D8000h~DBFFFh	18KB
SCSI型ハードディスク	DC000h~DCFFFh	4KB

■隠しオプション/MOVEHDBIOSの効力

p.201

/MOVEHDBIOSはマニュアルに記載されていない隠しオプションです。EMM386.EXEに/MOVEHDBIOSオプションを付けるだけで、SCSI/SASI型ハードディスクのBIOS ROMを、テキストVRAM領域の未使用部分（A5000h以降）に移動できます。

☆SCSIとIDEを接続している場合のUMB

ただし98MATE/FELLOWに内蔵するIDE型ハードディスクのBIOS ROM(18KB)は、/MOVEHDBIOSを使っても移動できません。また98NOTEに内蔵するハードディスクはカタログではSASI規格となっていますが、実際はIDE型ハードディスクをSASIにエミュレーション（疑似的にほかの規格に化けること）しています。したがって、/MOVEHDBIOSでハードディスクのBIOS ROMを移動できません。現在、IDE型ハードディスクのBIOS ROMを移動できるのはMemoryServer IIのみです。 p.121

なお、PC-9801DA/FAなど一部の機種では、/MOVEHDBIOSを付けただけでNEC版Windows3.1が起動できなくなります。98MATEでは問題なく起動できます。このように機種によっては不具合が出る場合がある点に、十分注意して使ってください。

■SCSIハードディスクROMの「からくり」

SCSI型のハードディスクBIOS ROMは、標準ではDC000h~DCFFFhまでの4KBを使用します。しかし、起動時にはDC000h~DDFFFhまでの8KBを占有しています。/MOVEHDBIOSを指定すると、DC000h~DCFFFhまでの4KBのBIOS ROMのみが移動し、DD000h~DDFFFhまでの4KB（未使用領域）は中途半端に残ってしまいます。

このままでは前ページの図のように、2番目と3番目のUMBブロックに分断されたままになってしまいます。UMBにプログラムをロードするときには、分断されたブロックのうち、最大サイズのブロックを優先して使います。したがって、UMBを確保するときにはできるだけ連続した領域になるように操作するのがポイントとなります。そこで次のように/Iオプションで、この未使用領域をUMB化することで、連続したUMBエリアにすることができます。

```
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB /MOVEHDBIOS /I=DC00-DFFF
```

/Iオプションの使い方は次ページ以降で詳しく説明します。なお、パソコン機種によっては、/MOVEHDBIOSで半端に残る4KBのROMをUMB化（/I）することができない場合があります。筆者のテストしたところでは、PC-9821As (98MATE) でこの現象が確認されました。逆にPC-9801DAでは、連続したUMBを確保できました。

未使用領域に UMB を割り当てる

PC-9821As の ROM に記録されている N88-BASIC 領域の一部（E8000h~F5FFFh の 57KB）を、強制的に UMB に設定しましょう。

操　作

CONFIG.SYS を次のように修正します（網掛け部分が追加した部分）。

修正前の CONFIG.SYS

```
FILES=30
BUFFERS=10
FCBS=1
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB /MOVEHDBIOS /I=DC00-DFFF
/I=E800-F5FF
DEVICE=A:¥DOS¥SMARTDRV.SYS 2048 128
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH,UMB
```

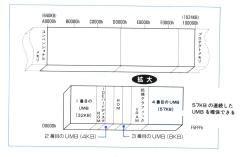
解　説

■確保できる UMB 容量

このCONFIG.SYS でパソコンを再起動すると、UMB の総容量は 100KB になります。また BASIC 領域に割り当てた UMB は、約 57KB の連続した領域になります。

(640KB)
A0000h　B0000h　C0000h　D0000h　E0000h　F0000h　(1024KB)
100000h
コンベンショナルメモリ
プロテクトメモリ
拡大
D0000h
1番目の UMB (32KB)
IDEハードディスクROM
ROM
拡張グラフィックVRAM
4番目の UMB (57KB)
F5FFFh
57KB の連続した UMB を確保できる
2番目の UMB (4KB)
3番目の UMB (8KB)

■隠しオプション/I の使用法

p.201

/I オプションは、指定したセグメントアドレス(4桁)範囲を強制的に UMB にするオプションです。アドレスとアドレスはハイフン「-」で区切ります。複数の領域を指定するには、次のように何個も/I を記述します。

```
EMM386.EXE /P=32 /UMB /DE00-DF00 /I=E800-F800 /I=A500-A700
```

なお、システム ROM/ハードディスク ROM/EMS ページフレームなどの重要なアドレスを/I で範囲指定してしまうと、ハングアップなど思わぬ障害が発生するので、使用には十分注意してください。/I はマニュアルに記載されていない隠しオプションです。したがって、NEC インフォメーションセンターに問い合わせても、サポートの対象外となります。

■機種によっては Windows が起動不可となる

PC-9801DA/FA では/I を付けた場合、NEC 版 Windows3.1 が起動できなくなる点に注意してください。一方、98MATE では/I=E800-FCFF (86KB) まで設定しても、問題なく起動できることを確認しています。

これは NEC 版 Windows3.1 が起動時に UMA 領域に「NEC」という ID があるかどうかをチェックし、発見できないと起動できないようにしているため

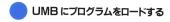
と思われます。このIDがBASIC領域とシステムBIOS ROMの中間付近に位置しているらしいことはわかっています（正確には不明）。このように、機種によっては不具合が出る場合がある点に十分注意して使ってください。なお、MELWAREとMemoryServer IIでは、このIDチェックをクリアするオプションがあります。p.229、212

■BASIC領域をUMB化するコツ

98シリーズでは全機種で共通して、UMA領域のE8000h以降の約64KBはN88-BASIC ROMが割り当てられています。このBASIC領域はBASICインタープリタとBASIC制御ルーチンの2つの領域に分けられます。ただしBASIC領域の詳細なアドレスや機能については非公開になっていて、しかも機種によって異なります。

したがって、使用する機器環境・ソフト環境に応じて、UMB化するBASIC領域の微調整が必要な場合があります。次の表はBASIC領域をUMB化するときの目安です。ご使用の環境で動作チェックをし、問題があれば次々にUMB化する範囲を狭めていくなどの調整を行うとよいでしょう。

/I指定	確保できる容量	補　足
/I=E800-FCFF	86KB	マニアックな人向けの領域
/I=E800-F8FF	69KB	〃
/I=E800-F800	65KB	〃
/I=E800-F5FF	57KB	DOSSHELLを使用する場合はこの指定にする
/I=E800-F400	49KB	PC-9801NS/T以降の機種ではこの指定にする
/I=E800-F31F	45KB	BASICインタープリタの領域
/I=E800-F300	44KB	もっとも安全で控えめな領域

UMB にプログラムをロードする

現在、約100KBのUMBが確保されています。CONFIG.SYSとAUTOEXEC.BAT の中からいくつかのプログラムを UMB にロードしましょう。

操　作

CONFIG.SYS と AUTOEXEC.BAT を次のように修正します（網掛け部分が修正個所）。

修正後の CONFIG.SYS

```
FILES=30
BUFFERS=10
FCBS=1
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB /MOVEHDBIOS /I=E800-F5FF
DEVICEHIGH =A:¥DOS¥SMARTDRV.SYS 2048 128  ←―― ここを修正
DEVICEHIGH =A:¥DOS¥KKCFUNC.SYS          ←―― ここを修正
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH, UMB
```

修正後の AUTOEXEC.BAT

```
@ECHO OFF
PATH A:¥DOS;A:¥
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
LH DOSKEY /INSERT   ←―― 追加
MOUSE /R
```

解　説

■UMBを活用する2つのコマンド

UMB領域を確保しただけの状態では何のメリットもありません。このUMB領域にプログラムをロードすることで、コンベンショナルメモリを節約できるようになります。UMBにプログラムをロードするには、次の2つのコマンドを使用します。

DEVICEHIGH

p.189

デバイスドライバをUMBにロードするためのCONFIG.SYSコマンドです。次の入力例では、PRINT.SYSをUMBにロードしています。なお、メモリマネージャ（MS-DOSではEMM386.EXE）や日本語入力プログラム（ATOK8やNECAIなど）は、DEVICEHIGH指定するとハングアップする場合があるので注意してください。

書式	:DEVICEHIGH=デバイスドライバ
入力例	:DEVICEHIGH=A:¥DOS¥PRINT.SYS /U

LOADHIGH（LH）

TSR（常駐型プログラム）をUMBにロードするためのMS-DOSコマンドです。LOADHIGHは略してLHと入力することもできます。次の入力例では、DOSKEYコマンドをUMBにロードしています。

書式	:LH TSR名
入力例	:LH DOSKEY /INSERT

■UMB活用の3法則

DEVICEHIGHやLHコマンドを実行したときに、UMBはどのように使われるのでしょうか。実はMS-DOSがUMBを使うときには、次の3つの法則があります。

最大サイズのUMBブロックから使う

UMBの総容量が100KBあるとします。通常はUMBは何個かのブロックにわかれています。このときMS-DOSは複数のメモリブロックの中から、最大サイズのものから優先して使いますから、次ページの図では4番目のUMBブロック（57KB）が最初に使われます。たとえばここに40KBのプログラム

をロードすると、残りは17KB (57KB-40KB=17KB) になります。したがって次にプログラムをロードするときには、1番目のUMBブロック (32KB) が使われることになります。

```
VMAP Version 2.01  Copyright (C) 1989-91 by c.mos
addr PSP  blks   size  owner/parameters            hooked vectors
---- ---- ----  ------  -------------------------   ---------------------------
D002-D7FE   1    32704  <free> ←――― 1番目のブロック
DC02-DCFE   1     4032  <free> ←――― 2番目のブロック
DE02-DFFE   1     8128  <free> ←――― 3番目のブロック
E802-F600   1    57312  <free> ←――― 4番目のブロック
                        --- UMB total:  100 KB ---
0586 sys    1     1808  himem                        DC
05F8 sys    1     7968  emm386                       1F 4B 67
07EB sys    4    33344  <config>
1013 <--    3     2960  command                      22 23 24 2E 2F
10CF-9FFF   1   586480  <free>
                                    ●
                                    ●
```

他のUMBブロックにまたがってロードできない

たとえば、30KBのデバイスドライバをDEVICEHIGHで指定したとします。このときUMBブロックの中に、30KB以上のUMBブロックがないと自動的にコンベンショナルメモリにロードされてしまいます。UMBの総容量が多くあっても、UMBブロックが十分なサイズでなければ意味がないのです。複数個の細切れのUMBブロックをつなぎ合わせてロードすることはできません。したがって、できるだけ連続したUMB(すなわち大きいサイズのUMBブロック) を確保することが重要になります。

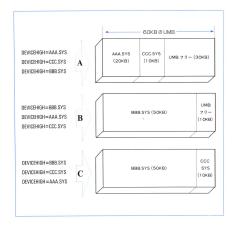
UMBブロック内では下位アドレスから使う

あるUMBブロックにプログラムをロードする場合は、下位のアドレス位置から使用されます。

■効率的なロード方法

前述した「UMB活用の3法則」を理解すると、プログラムをロードするときの順番が重要であることがわかります。たとえば、現在60KBの連続したUMBブロックが1個あるとします。ここに次のような3つのデバイスドライバをロードするときの、順番を考えてみてください。

```
DEVICEHIGH=AAA.SYS  ←  20KB
DEVICEHIGH=BBB.SYS  ←  50KB
DEVICEHIGH=CCC.SYS  ←  10KB
```

このケースでは次の［C］パターンの順序で DEVICEHIGH を記述すると、効率的に UMB を使うことができます。

便利な機能を付加する

ここでは MS-DOS の操作環境を快適にするための設定について解説します。

高速アクセス環境を設定する

ディスクアクセスをできるだけ高速に行えるよう設定しましょう。

操 作

CONFIG.SYS と AUTOEXEC.BAT を次のように修正します。

修正前の CONFIG.SYS

```
FILES=30
BUFFERS=10
FCBS=1
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB
DEVICE=A:¥DOS¥SMARTDRV.SYS 1024 128 /E
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH, UMB
```

☆修正後の CONFIG.SYS

```
FILES=30
BUFFERS=30          ← バッファ数を増やす
FCBS=1
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB
DEVICE=A:¥DOS¥SMARTDRV.SYS 3072 128     ← キャッシュ容量を増やす
DEVICE=A:¥DOS¥RAMDISK.SYS 1024     ← RAM ディスクを作る
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH, UMB
```

☆修正前の AUTOEXEC.BAT

```
@ECHO OFF
PATH A:¥DOS;A:¥
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
```

☆修正後の AUTOEXEC.BAT

```
@ECHO OFF
PATH A:¥DOS;A:¥
SET TEMP= D:¥          ← 作業領域を RAM ディスクに設定
SET DOSDIR=A:¥DOS
FASTOPEN A:=100        ← ディレクトリ情報のキャッシング
```

解 説

■高速化のための 3 つのポイント

ここでの作業は次の 3 点がポイントになります。

キャッシュバッファ容量を増やす

ディスクバッファ (BUFFERS) とディスクキャッシュ容量 (SMARTDRV.SYS) を増やすことで、ディスクに対する読み書きが高速になります。またSMARTDRV.SYS から/E を取ることで、EMS のページングを省略でき、高速になります。

一時作業領域を RAM ドライブにする

MS-DOS コマンドの一時作業領域 (テンポラリファイル) の作成場所をRAM ディスク (RAMDISK.SYS) に設定することで、MS-DOS コマンドのパイプ処理 (¦) /SORT コマンド/DOSSHELL の切り替えなどが高速になります。

ディレクトリ情報をキャッシングする

FASTOPN コマンドは指定ドライブのディレクトリ情報(ファイル名・ディレクトリ名・セクタの配置情報など) をメモリ上に記憶します。これによって目的のファイルに高速にアクセスできるようになります。

ワンタッチで高速リブートする

必要なときにワンタッチで高速にリセットできるように、リブートユーティリティ「HSB.EXE」をCONFIG.SYSに登録しましょう。

操　作

①HSB.EXEの入手

パソコン通信のホスト局（PC-VANやNIFTY-Serve）からHSB.EXEをダウンロードし、解凍（展開）します。ここでは解凍したHSB.EXEを、サブディレクトリ「A:¥UTLTY」にコピーします。

②CONFIG.SYSへの登録

CONFIG.SYSに次の1行を追加します。リセットボタンを押し、再起動すると、HSB.EXEがMS-DOSに認識されます。

```
DEVICE=A:¥UTLTY¥HSB.EXE VU X
```

③高速リブートの実行

MS-DOSが起動している状態で、CTRL＋GRPH＋DELキーを押します(MS-DOSのコマンドラインからHSB⏎と入力してもリブート可能)。

解　説

■HSB.EXEとは

HSB.EXE（作者:Masao氏）は、高速リブートユーティリティ（起動しているMS-DOSからメモリチェックや周辺機器チェックを省略して高速に再起動するプログラム）です。HSB.EXEは、PC-VANやNIFTY-Serveで無料で入手できるフリーソフトです。PC-VANでは98CLUB（J 98CLUB➡7➡2➡2➡＃172）に登録してあります。

また、ダウンロードしたファイル（HSB31EXE.LZH）はデータ圧縮した状態なので、元に戻す（解凍）にはアーカイバーソフト「LHA.EXE」（作者:吉崎栄泰氏）が別途必要です。LHAも、PC-VANやNIFTY-Serveで入手できるフリーソフトです。

ファイル削除トラブルを防ぐ

A ドライブと B ドライブのファイルを間違って削除してしまったときに復活できるよう、AUTOEXEC.BAT に MIRROR コマンドを登録しましょう。

操　作

AUTOEXEC.BAT を次のように修正します。

```
@ECHO OFF
PATH A:¥DOS;A:¥
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
MIRROR /TA /TB    ←―― この 1 行を追加
```

解　説

■MIRROR コマンドとは

MIRROR を実行するとメモリ上に常駐し、指定したドライブでのファイル削除を常に監視します。削除が実行されると指定ドライブに PCTRACKR.DEL というファイルに内容を記録します。このファイルは UNDELETE コマンドで削除ファイルを復活するときに使われます。たとえば C:¥ABC.TXT というファイルを削除してしまった後に復活させるには次のように操作します。

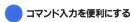
①「UNDELETE C:¥ABC.TXT ↵」と入力
②「復元しますか？」と表示されたら、Y を入力

ただし、MIRROR コマンドは必ずしも万能ではありません。削除後にひんぱんにディスクを読み書きすると復活できなくなる場合があるので、データのバックアップは常に必要です。

■MIRROR の常駐エリア

p.193

MIRROR は実行時に UMB に 12KB 以上の空きを見つけると、自動的に UMB に常駐する仕組みになっています。ただし、INSTALL (CONFIG コマンド) で MIRROR を指定するとエラーになります。INSTALL が強制的にコンベンショナルメモリにロードしようとするためです。

コマンド入力を便利にする

過去に入力したコマンドを後で呼び出して実行できるよう、DOSKEY コマンドを登録しましょう。

操　作

AUTOEXEC.BAT を次のように修正します。

```
@ECHO OFF
PATH A:¥DOS;A:¥
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
DOSKEY /INSERT     ←――― この 1 行を追加
```

解　説

■DOSKEY コマンドとは

コマンドラインからのコマンド入力機能は大変不便です。そこで DOSKEY を組み込むことによって、次のような便利な使い方ができるようになります。

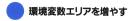
入力途中のコマンドを編集する

入力中のコマンドの任意の位置にカーソルを移動し、編集できます。カーソル移動は次のキー操作で行います。

キー操作	カーソルの移動先
SHIFT + ←	コマンドの先頭に移動
SHIFT + →	コマンドの末尾に移動
→	1文字分だけ左に移動
←	1文字分だけ右に移動

過去のコマンド履歴を表示・再利用する

過去に入力したコマンドを利用するには、次のキーで行います。

キー操作	機　能
↑	コマンド履歴を1つずつさかのぼって表示する
↓	コマンド履歴を古い順に1つずつに表示する
ROLL UP	コマンド履歴のもっとも古いものを表示する
ROLL DOWN	コマンド履歴のもっとも新しいものを表示する
f･7	これまで入力したコマンド履歴一覧を表示する
f･8	入力した文字列を含むコマンドを履歴から検索し表示する
f･9	コマンド履歴の中から指定した行番号のものを表示する

環境変数エリアを増やす

バッチファイルで環境変数を多用したいのですが、環境変数エリアが容量不足になりました。環境変数エリアを512バイトに拡大しましょう。

操　作

CONFIG.SYSを次のように修正します。

```
FILES=30
BUFFERS=10
FCBS=1
SHELL=A:¥COMMAND.COM /P /E:512   ←―― /Eオプションを追加
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB
DEVICE=A:¥DOS¥SMARTDRV.SYS 1024 128 /E
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH,UMB
```

解　説

■環境変数とは

環境変数とは、MS-DOS管理下で異なるプログラム(またはバッチプログラム)同士が連絡用に使う「掲示板」のようなものです。既定値では環境変数エリアは256バイトあります。当然、環境変数エリアを広げるとコンベンショナルメモリはその分だけ狭くなります。

なお、/Eを指定するときには必ず16の倍数で指定してください。環境変数は16バイト単位で増減するため、「/E:512」と指定しても「/E:527」と指定しても、確保される環境変数エリアは512バイトで、「/E:527」と指定した場合の残りの15KBは、未使用のまま無駄になってしまいます。

複数の FEP を切り替える(1)

一太郎 Ver.5 を使うときには「ATOK8」、桐を使うときには「松茸 Ver.3」、MS-DOS コマンドラインでは「AI かな漢字変換」を使いたいと思います。CONFIG.SYS とそれぞれの起動用バッチプログラムを作成しましょう。

操　作

①CONFIG.SYS への FEP 登録

CONFIG.SYS に次の 3 種類の FEP (日本語入力プログラム) を登録します。

☆CONFIG.SYS

```
FILES=30
BUFFERS=10
FCBS=1
SHELL=A:¥COMMAND.COM
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
DEVICE=A:¥M6¥MTTK3A.DRV A:¥M6 /ZA:¥M6/G/N/P「/P・/S1
DEVICE=A:¥M6¥MTTK3B.DRV
DEVICE=A:¥DOS¥NECAIK1.DRV
DEVICE=A:¥DOS¥NECAIK2.DRV A:NECAI.SYS
DOS=HIGH,UMB
```


②一太郎 Ver.5 の起動バッチ作成

次のような一太郎 Ver.5 起動用のバッチファイルを作成します。

☆TARO.BAT

```
@ECHO OFF
SELKKC 1      ← ATOK8 の選択
JXW           ← 一太郎 Ver.5 の起動
SELKKC 3      ← AI かな漢字変換の選択
```

③桐の起動バッチ作成

次のような桐 Ver.5 起動用のバッチファイルを作成します。

✿KIRI5.BAT

```
@ECHO OFF
SELKKC 2        ←―― 松茸 Ver.3 の選択
CD A:¥KIRIV5
kiri            ←―― 桐の起動
CD A:¥
SELKKC 3        ←―― AI かな漢字変換の選択
```

解 説

■複数の FEP の登録 p.105

MS-DOS Ver.5.0 対応の FEP であれば、CONFIG.SYS に複数種類を登録できます。ただし、その分だけコンベンショナルメモリや EMS メモリは狭くなります。また、複数の FEP を登録する際には、FEP 行の先頭に KKCFUNC.SYS (FEP 監視プログラム) を登録する必要があります。

■FEP の選択方法

複数登録してある FEP から、目的の FEP を選択するには SELKKC コマンドを使います。通常は「SELKKC 2⏎」のように FEP 番号を指定します。FEP 番号は CONFIG.SYS に登録した順に割り振られます。割り振られている FEP 番号を確認するには、「SELKKC /S⏎」と入力します。

```
A:¥>SELKKC /S

1      日本語変換システムATOK8 Ver 1.00
2      松茸 Ver3.71
3      NECAI Ver1.00

現在活動可能な 'かな漢' は 1 です.

A:¥>
```

複数のFEPを切り替える(2)

一太郎Ver.5起動時に「ATOK8」を組み込み、終了と同時にATOK8をメモリ上から解放したいと思います。一太郎起動用バッチプログラム「TARO.BAT」と、定義ファイル「ATOK8.FEP」を作成しましょう。

操　作

①定義ファイルの作成

次のような定義ファイル「ATOK8.FEP」をサブディレクトリ「A:¥ATOK8」にエディタで作成します。

ATOK8.FEP

```
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
```

②起動用バッチの作成

次のような起動用バッチプログラムをエディタで作成します。

TARO.BAT

```
@ECHO OFF
ADDDRV A:¥ATOK8¥ATOK8.FEP    ← ATOK8 の組み込み
JXW                          ← 一太郎 Ver.5 の起動
DELDRV                       ← ATOK8 の解除
```

解　説

■ADDDRV/DELDRV使用時の注意点

ADDDRVで組み込めるのはキャラクタ系デバイスドライバ(1文字単位でデータを送るタイプ:MOUSE.SYS/RSDRV.SYSなど)だけです。ブロック系デバイスドライバ(まとまった単位でデータを送るタイプ:EMM386.EXE/RAMDRV.SYSなど)は使用できません。また、1つのバッチファイルの中で2回以上連続してADDDRV/DELDRVを使うとメモリが解放されずに、フリーエリアが狭くなる場合があります。

AUTOEXEC.BAT 実行時の表示を消す

AUTOEXEC.BATの1行目に「ECHO OFF」を登録しているのに、「ECHO OFF」という文字と、起動時のMIRRORコマンドのメッセージが表示されます。これらがすべて表示されないように設定しましょう。

☆AUTOEXEC.BAT 実行時に表示されるメッセージ

```
Command バージョン 5.00A

A>ECHO OFF                        ← ECHO コマンドの表示
削除追跡ソフトウェアを組み込みます.

下記のドライブをサポートしています:
ドライブ A - 既定ファイルをセーブしました.     MIRROR コマンドの表示
ドライブ B - 既定ファイルをセーブしました.
ドライブ C - 既定ファイルをセーブしました.

組み込みは完了しました.
A:¥>
```

操　作

AUTOEXEC.BATを次のように修正します。

☆修正前の AUTOEXEC.BAT

```
ECHO OFF
PATH A:¥DOS;A:¥
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
MIRROR /TA /TB /TC
```

☆修正後の AUTOEXEC.BAT

```
@ECHO OFF        ← 先頭に@をつける
PATH A:¥DOS;A:¥
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
MIRROR /TA /TB /TC > NUL   ← NUL デバイスを指定する
```

解 説

■補助コマンド「@」とは

「ECHO OFF」を指定していないバッチプログラムでは、現在実行中の命令行が画面に逐一表示されます。「ECHO OFF」を指定すると、実行中の命令行は表示されなくなります。ただし、既定値は「ECHO ON」なので、最初の「ECHO OFF」という命令行だけは表示されてしまいます。

そこで、MS-DOS Ver.5.0から用意された補助コマンドに「@」があります。バッチファイルのコマンドの先頭に「@」を付けると、その行のみを画面に表示しなくなります。一般的には「@ECHO OFF」という使い方をしますが、他のコマンドの先頭に付けてもかまいません。

■NULデバイスの活用

MS-DOSでは「NUL」という英字には、ダミーデバイスが割り当てられています。ダミーデバイスとは「ゴミ箱」「ブラックホール」のようなもので、主に次の使い方があります。

コピーした結果を捨てる

たとえばディスクキャッシュに辞書ファイルを読み込ませるには、次のように入力します。

```
COPY ATOK8.DIC NUL↵
```

これでディスクキャッシュをロックすれば、高速に日本語入力ができるようになります。このときNULにコピーしたはずのATOK8.DICは、どこにも書き込まれていません。ブラックホールにコピー結果を捨てたことになります。

コマンドからのメッセージを非表示にする

コマンド実行時に「○○個のファイルをコピーしました」などというメッセージ表示を消すには、次のようにリダイレクト機能（〉記号）を使ってNULデバイスを指定します。

```
COPY ATOK8.DIC B: 〉 NUL↵
```

複数のCONFIG.SYSを切り替える

Windows3.1や一太郎Ver.5を起動するときには、それぞれ専用のCONFIG.SYSとAUTOEXEC.BATで再起動したいと思います。MS-DOSの持つリブートコマンド「CHGEV」を使って、再起動用のメニューバッチプログラムを作成しましょう。

操 作

①CONFIG.SYSとAUTOEXEC.BATの作成

- 「A:¥WINDOWS」にWindows起動用のCONFIG.SYSとAUTOEXEC.BATを作成します。
- 「A:¥TARO5」に一太郎起動用のCONFIG.SYSとAUTOEXEC.BATを作成します。

※このとき各AUTOEXEC.BATには目的のアプリケーションを起動するコマンドを記述してください。またAUTOEXEC.BATの末尾には「CHGEV /S」を記述してください

②メニューバッチプログラムの作成

エディタで次のようなバッチプログラム「XMENU.BAT」を作成します。

☆XMENU.BAT

```
@ECHO OFF
CLS
:選択
ECHO¥
ECHO   ◆◆◆ リブートメニュー ◆◆◆
ECHO¥
ECHO    «1» 一太郎 Ver.5
ECHO    «2» Windows3.1
ECHO    «0» 終了
ECHO¥
ECHO   ================================
ECHO¥
ECHO¥
BATKEY 0 番号を入力してください 》
IF ERRORLEVEL 3 GOTO 選択
IF ERRORLEVEL 2 GOTO Windows
IF ERRORLEVEL 1 GOTO 一太郎
IF ERRORLEVEL 0 GOTO 終了
:WINDOWS
CHGEV A:¥WINDOWS
GOTO 選択
:一太郎
CHGEV A:¥TARO5
GOTO 選択
:終了
EXIT
```

解　説

■CHGEV の隠しオプション

CHGEV はリブート用のコマンドです。通常は DOSSHELL からのみ使うサブコマンドです。隠しオプションとして、次の 4 つがあります。

オプション	機　能
/N	ノーマルモードに切り替えてリブートする
/H	ハイレゾリューションモードに切り替えてリブートする
/S	CONFIG.SYS と AUTOEXEC.BAT を元に戻してリブートする
/M	ハードディスクの起動メニューを表示する

「CHGEV A:¥WINDOWS⏎」などと入力すると、指定したパスにあるCONFIG.SYSとAUTOEXEC.BATでリブートします。このとき、起動ドライブのルートディレクトリに元からあったCONFIG.SYSとAUTOEXEC.BATは拡張子「CHG」にリネームされ、代わりに指定したパス上のCONFIG.SYSとAUTOEXEC.BATが複写されています。元の状態に戻すには「CHGEV /S⏎」を実行します。

■HSBを使って高速リブートする

p.99

CHGEVはリセットに時間がかかるという欠点があります。99ページで紹介したHSB.EXEを使って高速リブートするには、次の手順で操作します。

①CONFIG.SYSへ「DEVICE=HSB.EXE V Y-」という1行を追加します。

②サブディレクトリ「A:¥ENV」に次のようなファイルを作成します。

```
CONFIG.WIN     ←―― Windows起動用のCONFIG.SYS
AUTOEXEC.WIN   ←―― Windows起動用のAUTOEXEC.BAT
CONFIG.JXW     ←―― 一太郎起動用のCONFIG.SYS
AUTOEXEC.JXW   ←―― 一太郎起動用のAUTOEXEC.BAT
```

※ファイル拡張子にそれぞれ任意の英字を指定する点がポイントです

③AUTOEXEC.BATに次の2行を追加します。

```
SET HSBEXT=AAA  ←―― 再起動の際に一時的に作るファイルの拡張子を指定
SET HSBDIR=ENV  ←―― ファイルが格納されているディレクトリを指定
```

④一太郎を起動するには「HSB A:JXW⏎」と入力します。Windowsを起動するには「HSB A:WIN⏎」と入力します。このようにHSBの後に、ドライブ名と拡張子を指定する点がポイントです。この起動コマンドを前述の起動用メニューバッチに組み込めば、超高速リブートメニューができあがります。

第２章
MemoryServerⅡを徹底活用する

ここでは、MemoryServerⅡを使って最適な動作環境を設定するまでを、徹底的に解説します

MemoryServerIIをインストールする

ここでは、MemoryServerIIをMS-DOSとWindowsにインストールするまでを解説します。

フルオートインストールを実行する

MemoryServerIIを「フルオートインストール」(簡易組込)でハードディスク(ここではAドライブ)にインストールしましょう。

操　作

①インストーラの起動

・フロッピーディスク装置にMemoryServerII実行用ディスクをセットします。

・リセットボタンを押します。

☆インストーラを起動する

②ドライバのインストール

・メインメニューから「ドライバのインストール」にカーソルを移動し、↵キーを押します。

・接続されているディスク装置一覧が表示されるので、MemoryServer IIをインストールするディスク（ここではIDEハードディスク　#1）にカーソルを移動し、↵キーを押します。

・起動可能なパーティションが表示されるので、目的の領域（ここでは①:MS-DOS Ver.5.0）にカーソルを移動し、↵キーを押します。

・「ドライバのインストールを実行します」と表示されるので、↵キーを押します。

③インストール方法の選択

パソコンが再起動し、インストール方法の選択画面が表示されます。ここでは「フルオートインストール」にカーソルを移動し、↵キーを押します。

フルオートインストールを選択する

④インストールの実行

・ハードウェア環境・ソフトウェア環境の解析が実行され、最後にメモリの割り振りが自動的に行われます。このまま↵キーを押すと、MemoryServer IIのシステムファイルがハードディスクに転送されます。

・インストールが終了したら、フロッピーディスクを抜いてリセットボタンを押します。

解　説

■2つのインストール方法

MemoryServer IIでは、次の2種類のインストール方法が用意されています。どちらを選んでも、詳細な設定を行うには後でCONFIG.SYSの修正が必要になるので、操作例ではフルオートインストールを選択しています。

フルオートインストール

現在の環境（CPUの種類・MS-DOSバージョン・Windowsの有無・プロテクトメモリの容量など）を解析し、ファイル転送と最適なCONFIG.SYSの自動作成を行います。

カスタムインストール

メモリマネージャの詳細なセッティングを行いながら、ファイル転送とCONFIG.SYS作成を行います。

■自動作成されるCONFIG.SYS

現在、PC-9821As(i80486DX、33MHz)には14MBのプロテクトメモリ、IDE型ハードディスク（240MB）とSCSI型ハードディスク（270MB）が装着されています。そしてAドライブにはWindows3.1がインストールされています。この状態でフルオートインストールを実行すると、次のようなCONFIG.SYSが作成されます。

☆CONFIG.SYS

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /I /U ←── VCPI 対応のメモリマネージャ
DEVICE=A:¥MDEV¥IOSPRO¥DC10.EXE 3072 /W=2048 ←── ディスクキャッシュ
DEVICE=A:¥MDEV¥IOSPRO¥IOS10.EXE 1536 /X ←── RAM ディスク
BUFFERS=20
FILES=30
FCBS=1
SHELL=¥COMMAND.COM /P
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
DOS=HIGH, UMB
```

☆AUTOEXEC.BAT

```
A:¥MDEV¥IOSPRO¥DPMI32.EXE ←── DPMI サーバの起動
@ECHO OFF
PATH A:¥DOS;A:¥; A:¥MDEV¥IOSPRO ←── パスの追加
SET TEMP=A:¥DOS
SET DOSDIR=A:¥DOS
```

なお、CONFIG.SYS にあらかじめ登録してあるメモリドライバは、無効になります。たとえば、EMM386.EXE と SMARTDRV.SYS が登録してあるとします。フルオートインストールを実行すると、EMM386.EXE には REM が付けられて、注釈行になります。そして SMARTDRV.SYS は削除されます。

☆実行前

```
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB
DEVICE=A:¥DOS¥SMARTDRV.SYS 2048 128
```

☆実行後

```
REM DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB ←── REMで注釈行になる
```

ユーティリティをWindowsにインストールする

MemoryServer IIはWindows上からディスクキャッシュやRAMディスクをコントロールできます。ここではそのためのユーティリティ（デバイスコントロール）を、Windows3.1にインストールしましょう。

操　作

①インストーラの起動

・Windows3.1を起動します。
・フロッピーディスク装置（ここではCドライブ）にMemoryServer IIディスクをセットします。
・プログラムマネージャから［アイコン（F）］→［ファイル名を指定して実行（R）］を選択します。
・「C:¥IOSPRO¥WSETUP.EXE」と入力し、［OK］ボタンを左クリックします。

インストーラのファイル名を指定する

②ユーティリティの転送

・インストール先を指定するウィンドウが表示されます。ここではそのまま［インストール開始］ボタンを左クリックします。
・インストールの終了メッセージが表示されたら、［OK］ボタンを左クリックします。

解　説

■デバイスコントロールとは

p.138

この操作を実行すると、Windows に「MemoryServer II」というグループが新たに作成されます。このグループに入っている「デバイスコントロール」アイコンを左クリックすると、次のようなウィンドウが起動します。

☆デバイスコントロールに含まれるユーティリティ

デバイスコントロールでは、次のユーティリティを選択できます。

ユーティリティ	機　能
IOSIO の設定	RAM ディスクの容量を変更する
DCIO の設定	ディスクキャッシュの容量を変更する
解像度の切り替え	GA-1024A/1280A の表示解像度を変更する

UMBを大胆に使う

MemoryServer IIはSASI/SCSI/IDEの3タイプのハードディスクBIOS ROMの移動に対応するなど、UMBの活用という点ではもっとも優れたメモリマネージャです。

MemoryServer IIでUMBを活用するには、次の2つの方法が選択できます。

ユーティリティ「OPTUMB」を使う（方法1）

MemoryServer II付属のOPTUMB.EXEは、UMBの設定と設定されたUMBへのプログラムのロードを行うユーティリティです。OPTUMBはハード環境やソフト環境を自動認識した上で作業を行うので、指定してはならない領域をUMB化するなどのトラブルを自動的に避けてくれます。デバイスドライバのリファレンスと「にらめっこ」せずとも、高度なUMB設定ができます。UMB設定に関しては全面的にOPTUMBを使った方が便利です。

エディタでCONFIG.SYSを編集する（方法2）

エディタからCONFIG.SYSやAUTOEXEC.BATを編集するオーソドックスな方法です。OPTUMBではフォローしきれない細かな設定を短時間に行える点がメリットです。ただし、ハード環境やソフト環境による設定の違いを考慮しなければならず、リスクも大きいと言えるでしょう。

ここでは「方法1」をメインに解説します。

使用可能な UMB を調べる

現在、PC-9821As(i80486DX、33MHz)には14MBのプロテクトメモリ、IDE型ハードディスク(240MB)とSCSI型ハードディスク(270MB)が装着されています。このときのUMBの状態を調べましょう。

操　作

①UMBの容量表示

MS-DOSのコマンドラインから「MSTAT /U⏎」と入力します。

UMBの容量・アドレスを表示する

```
A:¥>MSTAT /U
MSTAT  Memory Status  Version 2.12
Copyright (C) 1992-1994 I・O DATA DEVICE Inc. All rights reserved.
■UMB情報
  アドレス        サイズ        ID      名前
  --------------------------------------------
  D001～D7FD    32720 07FD0h  ----  未使用  ←── 1番目のUMBブロック
  DD01～DFFF    12272 02FF0h  ----  未使用  ←── 2番目のUMBブロック

  全メモリ ........................... 44992 bytes  ←── UMBの総容量
  使用可能メモリの合計 ............... 44992 bytes
  使用可能メモリの最大 ............... 32720 bytes
A:¥>
```

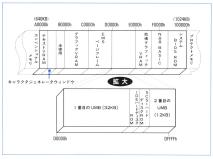
②UMBの内容表示

- 「INSPECT⏎」と入力します(ユーティリティの起動)。
- 「[M]メモリ」➡「UMB」➡「[M]メモリマップ」を選択します。
- 終了するには、ESC ➡ ESC ➡ ESC ➡ f·4 を押します。

※MSTAT、INSPECTは、MemoryServerIIに付属するメモリ表示ユーティリティです

解　説

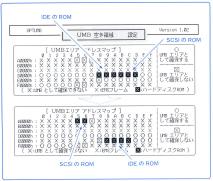
UMB の分断状況

この結果を見ると、UMB の総容量は約 44KB ありますが、2 つのブロック (32KB と 12KB の 2 ブロック) に分断されているのがわかります。

OPTUMBを使ってUMBを設定する

SCSI型ハードディスクとIDE型ハードディスクのBIOS ROMを次のアドレスに移動し、D0010h~DFFFFhまでの約64KBを連続したUMBブロックにしましょう。ここではUMB活用ユーティリティ「OPTUMB」を使います。

ROM種類	移動元アドレス	移動先アドレス
SCSI-HDD	DC000h~DCFFFh	A5000h~A6FFFh
IDE-HDD	D8000h~DBFFFh	E9000h~ECFFFh

SCSIとIDEのBIOS ROMを移動する

①OPTUMB の起動

・コマンドラインから「OPTUMB ⏎」と入力します。

・機能説明が表示されるので、⏎ キーを押します。

・次の図のようなメニューが表示されるので、「UMB 空き領域設定」にカーソルを移動し、⏎ キーを押します。

「UMB 空き領域設定」を選択する

②UMB の自動チェック実行

・修正対象となる CONFIG.SYS と AUTOEXEC.BAT があるドライブが表示されます。ここでは「はい（次に進む）」にカーソルを移動し、⏎ キーを押します。

・「再起動を行いますがよろしいですか」と表示されたら、「はい（再起動を行う）」にカーソルを移動し、⏎ キーを押します。

・「何かキーを押してください」と表示されたら、⏎ キーを押します。

・確認画面が表示されたら、⏎ キーを押します。

※これで MS-DOS はリセットがかかり、再び OPTUMB が起動します。もしメニュープログラムなどが自動起動するようになっていたら、それらを終了して MS-DOS プロンプトを表示してください

③Windows 使用の設定

「Windows を使用しますか？」と表示されます。この CONFIG.SYS 環境

下でWindowsを使用するなら、「はい（Windowsを使用する）」にカーソルを移動し、↵キーを押します。

④IDE 型ハードディスク ROM の移動

- UMBマップが表示されたら、「領域移動」にカーソルを移動し、↵キーを押します。
- 「EMSフレームを移動します」と表示されたら、↵キーを2回押します（これでEMSページフレームを移動しないことになる）。
- 「ハードディスクのBIOS ROMを移動します」と表示されたら、↵キーを押します。
- 点滅している4つの☆記号がIDEハードディスクのBIOS ROMです。→キーを1回押すと、IDEハードディスクのBIOS ROMはE9000h~ECFFFhに移動します。
- ↵キーを押します（これで移動が終了する）。

☆IDE 型ハードディスクの BIOS ROM を移動する

⑤SCSI 型ハードディスク ROM の移動

- 次に点滅している2つの☆記号が、SCSIハードディスクのBIOS ROMです。←キーを14回押すと、SCSIハードディスクのBIOS ROMはA5000h~A6FFFhに移動します。
- ↵キーを押します（これで移動が終了する）。

[UMBエリア アドレスマップ]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0000h :	×	×	×	×	×	☒	☒	☒	×	×	×	×	×	×	×	×
B0000h :	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
C0000h :	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
D0000h :	○	○	○	○	○	○	○	○	○	○	○	○	★	★	○	○
E0000h :	×	×	×	×	×	×	×	×	☒	■	■	■	■	○	○	○
F0000h :	○	○	○	○	○	☒	☒	×	×	×	×	×	×	×	×	×

(×:UMB として確保できない　×:EMSフレーム　■:ハードディスクROM)

○ UMB エリアとして確保する

☒ UMB エリアとして確保しない

[UMBエリア アドレスマップ]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0000h :	×	×	×	×	×	★	★	☒	×	×	×	×	×	×	×	×
B0000h :	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
C0000h :	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
D0000h :	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
E0000h :	×	×	×	×	×	×	×	×	☒	■	■	■	■	○	○	○
F0000h :	○	○	○	○	○	☒	☒	×	×	×	×	×	×	×	×	×

(×:UMB として確保できない　×:EMSフレーム　■:ハードディスクROM)

○ UMB エリアとして確保する

☒ UMB エリアとして確保しない

⑥設定内容の登録

・VMM386.EXE のオプション（設定前と設定後の比較）が表示されます。OK なら「はい（設定を行う）」にカーソルを移動し、↵ キーを押します。

・「再起動を行いますので、何かキーを押してください」と表示されたら、↵ キーを押します。

・確認画面が表示されたら、↵ キーを押します。

■OPTUMB の特徴

OPTUMB は、UMB 有効活用のための 3 つの機能を提供します。

UMB 空き領域設定

UMA 領域（A0000h~FFFFFh までの 384KB）を検索し、未使用エリアを UMB に設定します。EMS ページフレームやハードディスク ROM の移動もできます。

UMB 最適化設定

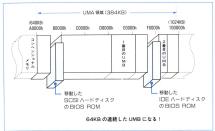
CONFIG.SYS/AUTOEXEC.BAT を解析し、もっとも効率的に UMB が使用されるようにロードする順番を設定します。

UMB 空き領域・最適化同時設定

「UMB 空き領域設定」と「UMB 最適化設定」を同時に行います。

■IDE の ROM 移動がもたらすメリット

p.86

この操作を実行すると、65KB (D0000h~DFFCFh) と 32KB (ED000h~F4FFFh) の2つのブロックからなる総容量 96KB の UMB が確保されます。総容量はそれほど多くないものの、IDE と SCSI を混在した環境で、UMB が細かく分断されていない点が優れています。また、プログラムのロードが簡単になります。

■OPTUMB が付加したオプション

操作例では OPTUMB を使って、CONFIG.SYS の VMM386.EXE のオプションを書き換えています。エディタで CONFIG.SYS を編集しても同じことですが、OPTUMB を使うと、マニュアルを見ずに複雑な設定ができるメリットがあります。ここでは次のような CONFIG.SYS ができあがります（網掛け部分が新たに追加されたオプション）。

```
BUFFERS=20
FILES=30
FCBS=1
SHELL=¥COMMAND.COM /P
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /I /U=D0-DF,ED-F4 /W=CC
/M=D8:16-E9;DC:8-A5 /NECID
DEVICE=A:¥MDEV¥IOSPRO¥DC10.EXE 3072 /W=2048
DEVICE=A:¥MDEV¥IOSPRO¥IOS10.EXE 1536 /X
```

```
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
DOS=HIGH, UMB
```

各オプションは次のような機能があります。

オプション	機　能
/I	メモリ設定内容を起動時に表示する
/U	指定した領域を UMB に設定する
/W	指定した領域を強制的に EMS ページフレームに設定する
/M	ハードディスク BIOS ROM を指定した領域に移動する
/NECID	N88-BASIC の ROM 領域を UMB にしたとき、NEC 版 Windows が起動できるようにする（NEC の ID チェック対策）

※オプションの詳しい機能については「リファレンス編」を参照してください。 p.229

■BASIC 領域を UMB に設定する p.88

98 シリーズの E8000h 以降にある N88-BASIC の ROM は、UMB の対象としては魅力的な領域です。たとえば E8000h~F5000h を UMB に指定するには、次のように記述します。

```
/U=E8-F5
```

なお、OPTUMB で「Windows を使用する」に設定すると、98 シリーズの E8000h 以降にある N88-BASIC の先頭の 4KB（E8000h~E8FFFh）が、UMB に設定できないようになります。MemoryServer II では E8000h にハードディスク ROM を移動したり、UMB を割り付けたりすると、Windows 起動時にハングアップします。したがって、Windows を使用するなら次のように記述します。

```
/U=E9-F5
```

ただし PC-9801DA では/U=E8-F6 まで設定しても、Windows3.1 を正常に起動できます。このように機種によっては、OPTUMB の設定を変更してもかまわない場合があります。

UMBにプログラムをロードする

総容量96KBのUMBに、プログラムをロードしたいと思います。OPTUMBを使ってロードしましょう。

操　作

①OPTUMBの起動

- コマンドラインから「OPTUMB⏎」と入力します。
- 機能説明が表示されるので、⏎キーを押します。
- 「UMB最適化設定」にカーソルを移動し、⏎キーを押します。
- 修正対象となるCONFIG.SYSとAUTOEXEC.BATがあるドライブが表示されます。ここでは「はい（次に進む）」にカーソルを移動し、⏎キーを押します。
- 確認画面が表示されたら、⏎キーを押します。

②ロードするデバイスドライバの指定

- 行番号6のDC10.EXEにカーソルを移動し、スペースキーを押します（選択されると先頭に「*」が付き、行が水色に反転する）。
- 同じ要領で行番号7（IOS10.EXE）と行番号8（KKCFUNC.SYS）を選択します。
- 選択が終了したら、⏎キーを押します。

☆スペースキーでドライバを選択する

③ロードする TSR（常駐プログラム）の指定

・行番号 11 の DOSKEY にカーソルを移動し、スペースキーを押します。

・同じ要領で行番号 12 の FASTOPEN を選択します。

・選択が終了したら、⏎ キーを押します。

スペースキーで TSR を選択する

④UMB 最適化の実行

・「再起動を行いますがよろしいですか」と表示されたら、「はい（再起動を行う）」にカーソルを移動し、⏎ キーを押します。

・確認のためのメッセージが4回表示されるので、それぞれ ⏎ キーを押します。

※これで MS-DOS はリセットがかかり、再び OPTUMB が起動します。もしメニュープログラムなどが自動起動するようになっていたら、それらを終了して MS-DOS プロンプトを表示してください

⑤UMB へのロードチェック

・再起動の確認のためのメッセージが 3 回表示されるので、それぞれ ⏎ キーを押します（プログラムのロードチェックのために再びリセットがかかる）。

・デバイスドライバのロードチェックの結果が表示されます（正常にロードできる行は先頭に「OK」が表示され、ロードできない行は「NG」が表示）。確認したら ⏎ キーを押します。

・TSR のロードチェックの結果が表示されます。確認したら ⏎ キーを押します。

⑥UMB の最適化の実行

・ロードチェック前後のメモリ状況が表示されたら、「はい（設定を行う）」にカーソルを合わせ、↵キーを押します。
・↵キーを押すと、再起動します。
・確認のためのメッセージが4回表示されるのでそれぞれ↵キーを押します。

解　説

■自動編集された CONFIG.SYS

この操作を実行すると、次のような CONFIG.SYS と AUTOEXEC.BAT が作成されます（網掛け部分が追加された記述）。

✪CONFIG.SYS

```
BUFFERS=10
FILES=35
FCBS=1
SHELL=A:¥COMMAND.COM /P
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /I /U=D0-DF,ED-F4 /W=CC
/M=D8:16-E9;DC:8-A5 /NECID
DEVICE= A:¥MDEV¥IOSPRO¥LUMB.EXE /M /B=1 A:¥MDEV¥IOSPRO¥DC10.EXE
3072 /W=2048 /S
DEVICE= A:¥MDEV¥IOSPRO¥LUMB.EXE /M /B=1 A:¥MDEV¥IOSPRO
¥IOS10.EXE 1536 /X
DEVICE= A:¥MDEV¥IOSPRO¥LUMB.EXE /M /B=1 A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
DOS=HIGH,UMB
```

☆AUTOEXEC.BAT

```
A:¥MDEV¥IOSPRO¥DPMI32.EXE
@ECHO OFF
PATH A:¥;A:¥DOS;A:¥BATCH;A:¥UTLTY;A:¥MDEV¥IOSPRO
prompt $p$g
SET DIRCMD=/O:-D
SET TEMP=B:¥
SET DOSDIR=B:¥
MIRROR /TA /TB /TC
A: ¥MDEV¥IOSPRO¥LUMB.EXE /M DOSKEY /INSERT
A: ¥MDEV¥IOSPRO¥LUMB.EXE /M FASTOPEN A: B:
```

■LUMB.EXE を使うメリット p.240

OPTUMB を使って UMB へプログラムをロードすると、LUMB.EXE が使われます。LUMB.EXE は、MS-DOS の DEVICEHIGH や LOADHIGH と同じです。しかし LUMB.EXE はデバイスドライバをロードする UMB ブロックを指定（/B）できる点が優れています。

例　`A:¥MDEV¥IOSPRO¥LUMB.EXE /M /B=1 A:¥DOS¥KKCFUNC.SYS`

なお、MS-DOS の DEVICEHIGH や LOADHIGH と LUMB.EXE は混在させてはいけません。どちらか一方に統一してください。

■再起動中のハングアップ　重要

OPTUMB はひんぱんにパソコンをリブート（再起動）させます。このとき途中でハングアップしてしまったら、SHIFT キーを押しながらリセットボタンを押してください。再び OPTUMB が起動するので、そこで UMB 最適化をやり直します。

目的の UMB ブロックにロードする

UMB に ATOK8B.SYS(約 15KB)をロードしたいと思います。現在、UMB は 2 つのブロック (30KB と 44KB) に分けられています。DEVICEHIGH コマンドでロードすると大きい方の UMB ブロック (44KB) にロードされます。しかし、ここでは小さい方の UMB ブロック (30KB) にロードしましょう。

操 作

①UMB ブロックの表示

「LUMB /I⏎」と入力します。ブロックの切れ目は空白行があります。ブロック番号は下位アドレスの方から 0~9 の範囲で付けます。したがって 30 KB の UMB ブロックは「ブロック番号 0」で、44KB の UMB ブロックは「ブロック番号 1」であることがわかります。

```
A:¥>LUMB /I
  アドレス       サイズ      ID     名前      割込みベクタ
D001～D0A9     2704 00A90h  D002  "hsb     "  21
D0AA～D7FD    30016 07540h  ----   未使用

E901～F4FF    49136 0BFF0h  ----   未使用

UMBの総容量        81856 バイト ( 79.9 KB)
空きUMBの合計      79152 バイト ( 77.2 KB)
空きUMBの最大      49136 バイト ( 47.9 KB)
システム設定       DOS=UMB
コンベンショナル・UMB間 切断
```


②CONFIG.SYS への登録

CONFIG.SYS の ATOK8 記述を次のように修正します。

修正前

```
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
```

☆修正後

```
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE= A:¥MDEV¥IOSPRO¥LUMB.EXE /M /B=0 A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
```

解　説

■ブロック選択のメリット

p.189

DEVICEHIGH コマンドを使うと、もっとも大きなUMBブロックから順に使われます（93ページのUMBの3法則を参照）。LUMB.EXEはロードするUMBブロックを選ぶことができるので、UMBを戦略的に活用できます。ちなみに、この操作を実行した後のUMBは次のようになります。

```
A:¥>LUMB /I
  アドレス        サイズ       ID      名前       割込みベクタ
 D001～D3D5   15696 03D50h  D002  "ATOK8B"     6F  ←―― ATOK8B.SYS
 D3D6～D47E    2704 00A90h  D3D7  "hsb     "   21
 D47F～D7FD   14320 037F0h  ----   未使用

 E901～F4FF   49136 0BFF0h  ----   未使用

 UMBの総容量        81856 バイト （ 79.9 KB)
 空きUMBの合計      63456 バイト （ 61.9 KB)
 空きUMBの最大      49136 バイト （ 47.9 KB)
 システム設定       DOS=UMB
 コンベンショナル・UMB間  切断
```

■FEPのUMBロード

注意

通常ATOK8や松茸などのFEP（日本語入力プログラム）は、UMBにロードすると、ハングアップするなどの現象が起きやすくなります。ただし、中にはロードしても問題がないと思われるドライバもあります。ATOK8の場合は、ATOK8A.SYS以外のドライバ（ATOK8B.SYS/ATOK8EX.SYS）はロードしても、特に問題は生じません。

ディスクバッファをUMBに移動する

現在、ディスクバッファは25個（BUFFERS=25）が指定され、約26KBのコンベンショナルメモリが消費されています。この内、24個をUMBに移動し、1個をコンベンショナルメモリに確保するように設定しましょう。

操　作

①CONFIG.SYSの修正

エディタにCONFIG.SYSを読み込み、BUFFERS指定を次のように修正します。

```
BUFFERS=25
  ↓
BUFFERS=1
```

②BEXコマンドの登録

エディタにAUTOEXEC.BATを読み込み、次の1行を追加します。

```
BEX 24 /H
```

解　説

■BEXコマンドとは

BEXはディスクバッファをUMBに作成するユーティリティです。通常はBUFFERSを1にして、BEXでそれより大きな個数を指定します。操作例を実行すると、コンベンショナルメモリは約24KB広くなります。

BEXは次のオプションが使用できます。

オプション	入力例	機　能
nn	BEX 25⏎	バッファ数を1~99の範囲で指定する
/H	BEX 25/H⏎	指定したバッファ数が確保できなかった場合に、可能な数だけUMBに確保する
/D	BEX /D⏎	UMB上のディスクバッファを削除する
/I	BEX /I⏎	現在のディスクバッファの状態を表示する
/Ann	BEX /A10⏎	指定したバッファ数（1~99）をUMBに追加する
/?	BEX /?⏎	ヘルプメッセージを表示する

便利な機能を使う

ここでは MemoryServer II の持つ便利な機能について説明します。

メモリをとことん節約する

UMB を活用する以外の方法で、コンベンショナルメモリをさらに節約できるように設定しましょう。

操　作

①ハンドル数の指定

VMM386.EXE を次のように修正します。

```
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /I /U /M /EH=42 /XH=16
```

②ディスクキャッシュの一部を EMS に移動

DC10.EXE を次のように修正します。

```
DEVICE=A:¥MDEV¥IOSPRO¥DC10.EXE 3072 /W=2048 /S
```

③DPMI の停止

DPIMI を使用しない場合は、AUTOEXEC.BAT から次の 1 行を削除します。

```
A:¥MDEV¥IOSPRO¥DPMI32.EXE
```

■メモリ節約のポイント

ここでの操作には、次の意味があります。

ハンドル数の変更

p.229

EMSやXMSにアクセスする際には、ハンドルという「整理券」のような役割をする番号を使います。あらかじめ何個までの整理券を用意するかを、ハンドル数で指定するわけです。ハンドル数が多いと、同時に多くのアプリケーションからのアクセス要求に応えられます。ただし、EMSやXMSはそれほど多数のアプリケーションが同時に使うわけではないので、多くのハンドルは空いたままになっています。

ハンドルはコンベンショナルメモリ上に作成されるので、この数値を少なくするとメモリ節約になります。この操作例では約128バイトが節約できます。/EHはEMSハンドル数、/XHはXMSハンドル数を指定します。

ディスクキャッシュの一部の移動

p.237

DC10.EXEを実行すると、データをキャッシュするバッファ領域と、DC10.EXE自体のプログラムがメモリに常駐します。バッファ領域は既定値ではEMB上に確保されます。DC10.EXE自身はコンベンショナルメモリ上に常駐します。

さらにDC10.EXEの常駐メモリは、詳しく見るとデータ部とプログラム部にわかれています。データ部はキャッシュサイズ32KBにつき16バイトが使われます。そこで/Sを付けることによって、データ部のみをEMS上に移動できます。ただし、データ部をEMSに移動すると、処理速度はやや低下します。

DPMIの停止

p.233

MemoryServer IIをフルオートインストールすると、AUTOEXEC.BATの先頭にDPMI32.EXE(DPMIサーバ)が登録されます。MemoryServer(旧バージョンの製品)ではVMM386.SYSに内包されていたDPMIが切り離されたものです。

DPMIサーバを必要とするアプリケーションは、まだ少ないのが現状です。また、Windows3.1はDPMIサーバを内蔵しているので、DPMI32.EXEを削除しても問題ありません。

※この他、メモリを節約するには「第1章 MS-DOS Ver.5.0Aを徹底活用する」(*p.75*) /BEXの活用 (*p.133*) /UMBの活用 (*p.118*) を参考にするとよいでしょう

DOSSHELL から一太郎を起動できるようにする

DOSSHELL から一太郎 Ver.5 が起動できません。これは DOSSHELL がプロテクトメモリをすべて EMB として予約してしまっているためです。そこで DOSSHELL 起動時には 4MB (4096KB) だけを確保し、残りのプロテクトメモリはフリーになるように設定しましょう。

操 作

CONFIG.SYS を次のように修正します（網掛け部分が追加した記述）。

```
BUFFERS=10
FILES=35
FCBS=1
SHELL=A:¥COMMAND.COM /P
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /I /U /X=4096 /T=A:¥DOS
¥EXTDSWAP.SYS
DEVICE=A:¥MDEV¥IOSPRO¥DC10.EXE 3072 /W=2048 /S
DEVICE=A:¥MDEV¥IOSPRO¥IOS10.EXE 1536 /X
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
DOS=HIGH, UMB
```

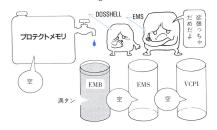
解 説

■起動ができない理由

DOSSHELL は起動時に空いているプロテクトメモリすべてを、XMS 規格が管理するメモリ「EMB」として予約してしまいます。

MemoryServer II はアプリケーションからの要求に応じて、プロテクトメモリを EMS/VCPI/XMS などに自動的に配分する便利な機能を持っています。ところが DOSSHELL のような欲張りなアプリケーションがあると、MemoryServer II のフレキシブルなメモリ配分が逆にアダとなって、EMS/VCPI/DPMI に割り当てるメモリがゼロになってしまいます。

このため、一太郎 Ver.5/松 Ver.6 など起動時に必ず EMS や VCPI を必要とするアプリケーションは、DOSSHELL から起動できなくなるわけです。

DOSSHELL がすべてのプロテクトメモリを独占する

■EMB の上限を定める/X オプション　重要

/X はあらかじめ EMB の上限を固定してしまう機能があります。/X=4096 と指定すると、MS-DOS は EMB の容量が 4MB であると判断します。これによって DOSSHELL 起動時には、EMB として予約されるのは 4MB だけになります。残りのプロテクトメモリは EMS や VCPI として、自由に使えるようになります。DOSSHELL を MemoryServer II で使う場合は必ず必要なオプションです。

■拡張タスクスワップをサポートする　p.229

/T は MS-DOS Ver.5.0A の拡張タスクスワップを利用するオプションです。必ず「EXTDSWAP.SYS」を指定します。なお、/T は EMB の上限を自動的に設定する機能もあります（/X と同じ機能）。ただし、/T は上限が大きいため（搭載するプロテクトメモリに近い容量）、やはり一太郎 Ver.5 が起動できないなど実質的な効果はありません。したがって、必ず/X の指定とペアで指定してください。

Windows 上から RAM ディスク容量を変更する

現在 RAM ディスク（IOS10.EXE）には 1536KB が設定されています。これを Windows 上から一時的に 2MB に変更しましょう。

操 作

①デバイスコントロールの起動

p.116

「MemoryServer II」グループから「デバイスコントロール」アイコン→「RAM ディスクコントロール」を選択します。

②RAM ディスク容量の変更

- ［RAM ディスクの容量（S）］には「2048」と入力し、［設定］を左クリックします。
- 「Windows を再起動しますか」と表示されるので、［はい（Y）］を左クリックします。

※データを保存していないアプリケーションがある場合は［いいえ（N）］を選択し、データを保存してからもう一度①からやり直してください。変更後の容量は電源を OFF（またはリセット）にするまで有効です

ディスクの性能を測定する

現在、Bドライブにはインターフェイスが SCSIIIタイプ、データ転送がバスマスタ方式のハードディスク（270MB）が接続されています。このディスクの性能を INSPECT コマンドでテストしましょう。

操　作

①ディスク性能測定の起動

- コマンドラインから「INSPECT ↵」と入力します（メニューの表示）。
- 「[B] 性能測定」➡「[D] ディスク」を左クリックします。

②ディスク性能テストの実行

- 「[D] ディスク」➡「[D] ドライブ変更」を左クリックします。
- 「B:　SCSI HDD」を選択し、[↵ 選択] を左クリックします。
- [↵ 性能測定] を左クリックします。

③性能測定の終了

- ESC ➡ ESC を選択します。
- f·4 キーを押します。

解　説

■INSPECTコマンドの用途

INSPECTコマンドを実行すると、次の図のようなベースメニューが表示されます。この中の「[M] メモリ」にあるUMBマップは、非常に使用価値が高い便利なユーティリティです。本書執筆でのUMB解析はこのユーティリティが役に立ちました。VMM386.EXE以外のメモリマネージャでも使うことができます。

☆INSPECTには便利なユーティリティがある

それぞれのボタンには次のようなユーティリティが用意されています。

ボタン	名称・機能	用　途
[O]	DOS	MS-DOSの一般的な情報（CONFIG.SYSの内容など）を表示する
[P]	プロセッサ	CPUまたは数値演算プロセッサの性能測定をする
[B]	性能測定	メモリ・ディスクなどの性能測定をする
[M]	メモリ	各種メモリ情報を表示する
[D]	ディスク	ディスクの接続状態やアクセス性能を表示する
[S]	システム	メモリスイッチの表示・変更を行う
[U]	補助機能	ファイルの検索・表示・表示形式変更を行う
[HELP]	説明表示	INSPECTの操作方法を表示する
[f·4]	終了	INSPECTを終了する

ここでは、32ビット用のメモリマネージャであるMELWARE for Windowsを使って最適な動作環境を設定するまでを、徹底的に解説します

MELWARE for Windows をインストールする

ここでは MELWARE for Windows をインストールするまでを解説します。

MELWARE for Windows を インストールする

MELWARE for Windows（以下 MELWARE for Win と記述）をハードディスク（ここでは A ドライブ）にインストールしましょう。なお、ここではインストール方法は「自動設定」を選択します。

①CONFIG.SYS のチェック/修正

・エディタで CONFIG.SYS を読み込みます。

・メモリマネージャをすべて削除し、次のような CONFIG.SYS にします。なお、FEP（日本語入力プログラム）などは行頭に REM 命令を挿入し、一時的に注釈文に設定しておくとよいでしょう。

・CONFIG.SYS を保存したらリセットボタンを押します。

☆インストール用の CONFIG.SYS

```
BUFFERS=20
FILES=30
FCBS=1
SHELL=A:¥COMMAND.COM /P
REM DEVICE=A:¥DOS¥KKCFUNC.SYS
REM DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
REM DEVICE=A:¥ATOK8¥ATOK8B.SYS
REM DEVICE=A:¥ATOK8¥ATOK8EX.SYS
```

REM を挿入

②インストーラのコピー

・フロッピーディスク装置（ここではBドライブ）に「MELWARE for Windows」ディスクをセットします。

・コマンドラインから次のように入力します。

```
COPY B:¥MELSETUP.* A:¥ ⏎
```

※3つのファイルがコピーされます

③インストーラの起動

・コマンドラインから次のように入力します。

```
MELSETUP ⏎
```

・「インストールを行いますか？」と表示されたら、Y を押します。

④自動設定の選択

・インストール先のディレクトリ名「A:¥MEL4WIN¥」が表示さます。ここではそのまま ⏎ キーを押します。

・メインメニューが表示されたら、「自動設定」にカーソルを合わせて ⏎ キーを押します。

☆メモリの配分をマニュアル設定する

```
MELWARE for WINDOWS Ver.1.1                    Copyright(C) 1994 MELCO,inc.

自動設定 <マニュアル設定>

◇RAMディスク                使用しない／[ 2048KB]
◆ディスクキャッシュ          使用しない／[ 3072KB]

◇UMB機能                    使用しない／使用する
 XMS+EMSメモリ                         [ 9216KB]

 *インストールドライブは A です。

   F1:設定例1,  F2:設定例2,  F3:設定例3,  F4:マニュアル設定

操作説明                      マニュアル設定
↑↓:項目移動                   ディスクキャッシュ(HYPERDSK.EXE)で使用する
←→:機能選択／容量調整         容量を指定します
↵ :登録実行
ESC :前画面／取消し

F10 :インストールドライブの変更
```

⑤メモリの配分

・f·4（マニュアル設定）キーを押します。

・「◇RAM ディスク」の容量が 2048KB と表示されるように、→または←キーを押します（増減は 128KB 単位、→は増、←は減）。

・↓を押し、カーソルを「◇ディスクキャッシュ」に移動します。

・「◇ディスクキャッシュ」の容量が 3072KB になるように→または←キーを押します。

・「◇UMB 機能」は既定値で「使用する」が選択されているので、ここではそのままにします。

・配分が決定したら、⏎キーを押します。

⑥インストールの実行

・「よろしいですか？」と表示されたら、Yを押します。

・次の図のようなメッセージが表示されるので、そのまま⏎キーを押します。

```
SYSTEM.INIの32BitDiskAccess=onをoffにします.
HYPERDSK使用時は「BIOSを使用する」必要があります.
SYSTEM.INIはSYSTEM.ORGに保存されます.
何かキーを押してください
```

・「MELWARE の登録が完了しました」と表示されたら、フロッピーディスクを抜いてからリセットボタンを押します。

解　説

■「MELWARE for Windows」と「MELWARE Ver.5」 重要

1994年9月現在、MELWAREのパッケージには、「MELWARE for Windows」と「MELWARE Ver.5」の2種類のディスクが入っています。前者にはWindows 環境をより意識した、高速高機能版のメモリマネージャが入っています。後者は旧来の 16 ビットマシンもその範疇に入れたメモリマネージャが入っています。

32 ビットマシンをお持ちのユーザーなら、Windows 使用のいかんにかかわらず、「MELWARE for Windows」を選択してください。本書では 32 ビットマシンを対象としているため、「MELWARE for Windows」を中心に解説しています。

■WINDOWS.INI 修正のメッセージについて

p.166

MELWARE for Win でディスクキャッシュ（HYPERDSK.EXE）を選択すると操作⑥のようなメッセージが出ます。HYPERDSK.EXE が、Windows の常設スワップファイルの 32 ビットアクセスに対応していないためです。

■2 つのインストール方法

MELWARE for Win では次の 2 種類のインストール方法が用意されています。どちらを選んでも、詳細な設定を行うには後で CONFIG.SYS の修正が必要になるので、操作例では「自動設定」を選択しています。

自動設定

現在の環境（CPU の種類・MS-DOS バージョン・Windows の有無・プロテクトメモリの容量など）を解析し、ファイル転送と最適な CONFIG.SYS の自動作成を行います。

オプション設定

メモリマネージャの詳細なセッティングを行いながら、ファイル転送と CONFIG.SYS 作成を行います。

■自動作成される CONFIG.SYS

自動設定を行うと、次のような CONFIG.SYS が自動作成されます。また操作①で REM コマンドを挿入しておいた FEP があれば、REM を削除しておきましょう。

作成される CONFIG.SYS

```
FILES=30
BUFFERS=20
DOS=HIGH,UMB
DEVICE=A:¥MEL4WIN¥MELEMM.386 /HM /M D0,D4,D8,DC
DEVICEHIGH=A:¥MEL4WIN¥HYPERDSK.EXE C:3072 CW:3072
DEVICEHIGH=A:¥MEL4WIN¥EXDISK.EXE X 2048
FCBS=1
SHELL=A:¥COMMAND.COM /P
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
```

Windowsユーティリティをインストールする

Windows 用のユーティリティをハードディスクにインストールしましょう。なお、この操作は、すでに Windows がインストールされている環境でのみ実行可能です。

インストールされた Windows ユーティリティ

操 作

①インストーラの起動

- フロッピーディスク装置（ここではＢドライブ）に「MELWARE for Windows」ディスクをセットします。
- コマンドラインから次のように入力します。

```
MELSETUP ↵
```

- 「インストールを行いますか？」と表示されたら、Y を押します。

②メニューの選択

- インストール先のディレクトリ名「A:￥MEL4WIN￥」が表示さます。ここではそのまま ↵ キーを押します。
- メインメニューが表示されたら、「Windows ユーティリティのインストール」にカーソルを移動し、↵ キーを押します。

③インストールの実行

・「よろしいですか？」と表示されたら、 Y を押します。
・Windows が起動し、インストール先を設定するウィンドウが表示されます。ここでは、そのまま［継続 (C) ］ボタンを左クリックします。
・「インストールが終了しました」と表示されたら、［OK］ボタンを左クリックします。

解 説

■MELWARE for Win のユーティリティとは

Windows ユーティリティには、次の 14 種類のユーティリティが含まれます。

名称・機能	用 途
リソースモニタ	Windows のリソースの使用状況を表示する
ディスクコピー	フロッピーディスク同士のコピーを行う
ディスクフォーマット	フロッピーディスクの初期化を行う
ファイルシュレッダ	不要なファイルをごみ箱に入れる（不可視属性）
ウィンドウズターミネータ	Windows の終了や再起動を制御する
ファイルクリーナー	ハードディスク上の不要ファイルの検索と削除をする
アイコンコントローラ	アイコンの表示を詳細にカスタマイズする
デバイスドライバモニタ	HYPERDSK と EXDISK の動作状況の表示と修正をする
ルーペ	画面上の指定領域を拡大/縮小表示する
クロック	画面上に時計を表示する
スクラッチパッド	作業中に思いついたアイデアをメモ書きする
ウォールマン	壁紙の表示と設定を行う
デスクスプレッド	仮想デスクトップを複数作成する
ミニランチャ	アプリケーション起動用のミニメニュー

UMBを大胆に使う

MELWARE for Winは、MemoryServer IIに比べるとUMB活用ではやや力不足ですが、「わかりやすさ」と「簡単操作」という点では、一番安心して使えるメモリマネージャです。

使用可能なUMBを調べる

現在、PC-9821As(i80486DX、33MHz)には14MBのプロテクトメモリ、IDE型ハードディスク（240MB）とSCSI型ハードディスク（270MB）が装着されています。このときのUMBの状態を調べましょう。

操　作

①メモリマップの表示

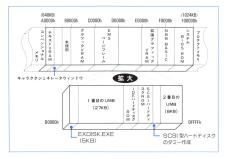
MS-DOSのコマンドラインから、次のように入力します。

```
MEL4WIN¥UMBSTAT /U ↵
```

```
A:¥>MEL4WIN¥UMBSTAT /U
MELWARE  MS-DOS & UMB CONTROLER
Rev 2.07  :  Oct . 06 1992  :  MELCO Inc. JAPAN

 M  MCB Addr Owner Para  Byte    Name
 - ---- ---- ----- ---- ------ ---------
 D D002 D003  D003 0144   5184 RESERVED
 M D147 D148  0000 06B6  27488   FREE
 M D7FE D7FF  0008 0602  24608 RESERVED
 Z DE01 DE02  0000 01FE   8160   FREE

A:¥>
```

EXDISK.EXE
1番目のUMBブロック（27KB）
ハードディスクのBIOS ROM（IDE＋SCSI）
2番目のUMBブロック（8KB）

UMBの分断状況

この結果を見ると、UMBの総容量は約40KBで、2つのブロック（27KBと8KBの2ブロック）に分断されているのがわかります。

■ユーティリティを使うためのパス設定 重要

MELWARE for Winをインストールすると、既定値ではサブディレクトリ「A:¥MEL4WIN」に格納されます。ユーティリティを簡単に使うためには、AUTOEXEC.BATのPATH設定にA:¥MEL4WINを追加してください。

例　PATH A:;A:DOS; A:¥MEL4WIN ↵

■UMBSTATとは

UMBSTATは、UMBやコンベンショナルメモリのメモリマップを表示するためのユーティリティです。TSR (常駐型プログラム) やデバイスドライバの常駐量のみを調べることもできるので、UMBへのロード (UMBLOADコマンド) のための常駐量チェックに使うこともできます。

オプション	機　能
/E	EMSメモリに関する情報を表示する
/L	UMBとコンベンショナルメモリのリンク・アンリンクを実行する
/M	コンベンショナルメモリのメモリマップを表示する
/U	UMBのメモリマップを表示する
/V	デバイスドライバとTSRの情報を表示する

ハードディスクROMを移動する

SCSI型ハードディスクのBIOS ROMをA5000h以降に移動させましょう。

操 作

CONFIG.SYSのMELEMM.386に次のオプションを追加します。

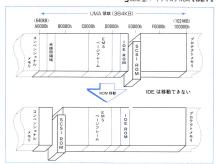
```
DEVICE = A:¥MEL4WIN¥MELEMM.386 /HM /M D0,D4,D8,DC /SW1
```

解 説

■ハードディスクROMの移動

p.86

/SW1はD0000h~DFFFFhの範囲に存在するハードディスクのBIOS ROMを、4KB単位でA5000h以降に移動します。ただし移動先の空き容量は12KB(A5000h~A7000h)なので、最大でも12KBまでのハードディスクROMしか移動できません。したがって、IDEのROM(16KB)は対象外になり、SCSIかSASIのみに限定されます。機種によっては/SW1を指定してもSCSIのROMが移動できないことがあります。筆者のテストではPC-9801DAでは移動でき、98MATEでは移動できませんでした。

☆SCSI型ハードディスクROMを移動する

ユーティリティでUMBを最適化する

UMB最適化ユーティリティ「オプティマイズ」を使って、デバイスドライバとTSR（常駐型プログラム）をUMBへ移動しましょう。

操　作

①インストーラの起動

・フロッピーディスク装置（ここではBドライブ）に「MELWARE for Windows」ディスクをセットします。

・コマンドラインから次のように入力します。

MELSETUP ↵

・「インストールを行いますか？」と表示されたら、Y を押します。

②メニューの選択

・インストール先のディレクトリ名「A:¥MEL4WIN¥」が表示されます。ここではそのまま ↵ キーを押します。

・メインメニューが表示されたら、「オプティマイズの実行」にカーソルを移動し、↵ キーを押します。

③ファイルのコピー

・処理についての説明が表示されたら、↵ キーを押します。

・「よろしいですか？」と表示されたら、Y を押します。

※OPTIMIZE.COMなどUMB最適化のためのプログラムがハードディスクにコピーされます

④UMB最適化の実行

・「フロッピーを取り出し、何かキーを押してください」と表示されます。「MELWARE for Windows」ディスクを取り出して、↵ キーを押します。

・「オプティマイズを開始します」と表示されたら、スペースキーを押します。

・「オプティマイズが終了しました」と表示されたら、スペースキーを押します。

```
＊＊＊　フロッピーディスクを取り出し，何かキーを押してください．　＊＊＊
+OPT_PHASE0_END----------------------------+
¦ｵﾌﾟﾃｨﾏｲｽﾞを開始します。                    ¦
¦中断したい時はESCを押してください。         ¦
¦ｵﾌﾟﾃｨﾏｲｽﾞを実行すると、このﾘｾｯﾄを含め      ¦
¦３回のﾘｾｯﾄを行います。                      ¦
¦ｽﾍﾟｰｽｷｰを押してください。                   ¦
+------------------------------------------+
```

解　説

■オプティマイズとは

「オプティマイズ」は、UMBを最大限に活用するためのUMB最適化ユーティリティです。オプティマイズは次のような処理を実行します。

①UMA領域を検索し、空いている領域をUMB化（/Mにアドレスを追加）する

②CONFIG.SYSとAUTOEXEC.BATに登録してあるプログラムの常駐量を調べ、どのプログラムをどのUMBブロックにロードすれば効率的かを決める

③プログラムがUMBへロードされるように、CONFIG.SYSとAUTOEXEC.BATを書き換える

■オプティマイズ使用時の注意点

オプティマイズ使用時には次の点に注意してください。

ディップスイッチSW2-5の変更要求が出る

ディップスイッチSW2-5（メモリスイッチの保持）がOFF（保持しない）になっていると、ONにするように要求するメッセージが表示され、オプティマイズは終了します。ディップスイッチのメニューは、HELPキーを押しながらリセットボタンを押すと表示できます。

リセットがかからない

オプティマイズは処理の中で、合計3回のリセットを行います。このとき環境によってはリセットがかからずに、そのまま画面が止まってしまうことがあります。この場合はパソコンのリセットボタンを押すと、処理を続行できます。

処理後のCONFIG.SYSの修正

オプティマイズが指定したプログラムは、必ずしも完全にUMBに入るとは限りません。また、MIRRORコマンドのように自動的にUMBへ移動するプログラムを指定してしまうこともあります。したがって、オプティマイズ実行後はCONFIG.SYSとAUTOEXEC.BATをよく点検して、必要に応じて修正するなどの作業が必要です。

■自動編集された CONFIG.SYS

この操作を実行すると、次のような CONFIG.SYS と AUTOEXEC.BAT が作成されます。エディタで CONFIG.SYS/AUTOEXEC.BAT を編集しても同じことですが、オプティマイズを使うとマニュアルを見ずに複雑な設定ができます。

☆実行前の CONFIG.SYS

```
FILES=30
BUFFERS=20
DOS=HIGH, UMB
DEVICE=A:¥MEL4WIN¥MELEMM.386 /HM /M D0, D4, D8, DC
DEVICEHIGH=A:¥MEL4WIN¥HYPERDSK.EXE C:3072 CW:3072
DEVICEHIGH=A:¥MEL4WIN¥EXDISK.EXE X 2048
FCBS=1
SHELL=A:¥COMMAND.COM /P
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
```

☆実行後の CONFIG.SYS

```
FCBS=1
SHELL=A:¥COMMAND.COM /P
FILES=30
BUFFERS=20
DOS=HIGH, UMB
DEVICE=A:¥MEL4WIN¥MELEMM.386 /SW1 /HM /M D0, D4, D8, DC, E8, EC, F0
/NECWIN
DEVICEHIGH=A:¥MEL4WIN¥HYPERDSK.EXE C:3072 CW:3072
DEVICE= A:¥MEL4WIN¥UMBLOAD.SYS /UA:4 A:¥MEL4WIN
¥EXDISK.EXE X 2048
DEVICE= A:¥MEL4WIN¥UMBLOAD.SYS /UA:4 A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE= A:¥MEL4WIN¥UMBLOAD.SYS /UA:4 A:¥ATOK8¥ATOK8B.SYS
DEVICE= A:¥MEL4WIN¥UMBLOAD.SYS /UA:4 A:¥ATOK8¥ATOK8EX.SYS
```

☆実行前の AUTOEXEC.BAT

```
@ECHO OFF
PATH A:¥;A:¥DOS;A:¥BATCH;A:¥UTLTY;A:¥WINDOWS;A:¥JUST5
;A:¥MEL4WIN
SET JW2P=A:¥GORO;A:¥HANA3;A:¥SANSIRO;A:¥TARO5;A:¥JW2
;A:¥JEDIT
PROMPT $p$g
SET DIRCMD=/O:-D
SET TEMP=A:¥
SET DOSDIR=A:¥DOS
MIRROR.COM
FASTOPEN A:
DOSKEY /INSERT
```

☆実行後の AUTOEXEC.BAT

```
@ECHO OFF
PATH A:¥;A:¥DOS;A:¥BATCH;A:¥UTLTY;A:¥WINDOWS;A:¥JUST5
;A:¥TAROWIN;A:¥JSLIB;A:¥MDEV¥IOSPRO;A:¥MEL4WIN
SET JW2P=A:¥GORO;A:¥HANA3;A:¥SANSIRO;A:¥TARO5;A:¥JW2
;A:¥JEDIT
PROMPT $p$g
SET DIRCMD=/O:-D
SET TEMP=B:¥
SET DOSDIR=A:¥DOS
A:¥MEL4WIN¥UMBLOAD.COM /UA:4 A:¥DOS¥MIRROR.COM
FASTOPEN A:
DOSKEY /INSERT
```

目的の UMB ブロックにロードする

UMB に DOSKEY コマンド（約 6KB）をロードしたいと思います。現在、UMB は 4 つのブロック（32KB/8KB/3KB/16KB）に分けられています。LOADHIGH コマンドでロードすると、大きい方の UMB ブロック（32KB）にロードされます。しかし、ここでは 16KB の方の UMB ブロックにロードしましょう。

操 作

①UMB ブロックの表示

「UMBLOAD⏎」と入力します。左端の「No」とあるブロック番号を見てください。目的のブロック番号は「4」であることがわかります。

```
A:¥>UMBLOAD
UMB LOADER Ver 2.07  Copyright (C) 1994 by MELCO,Inc.Japan

[書式]    d>UMBLOAD </UA:n> filename.COM or EXE
[オプション] /UA:n  常駐領域を指定出来ます （省略可能）
               +-> n=使用可能なUMB領域の番号

 No   UMB Area    Size      Status
---- ----------- ------ ----------------
  1  D002 - D7FE  32704 使用可能
  2  DE02 - DFFE   8128 使用可能
  3  E802 - E8DC   3488 使用可能
  4  E8E2 - EA26   5184 Used [EXDISK  ]
  4  EA27 - EB2A   4144 Used [KKCFUNC ]
  4  EB2B - EF00  15696 Used [ATOK    ]
  4  EF01 - EFDC   3504 Used [ATOK    ]
  4  EFDD - F400  16944 使用可能

使用可能なメインメモリのサイズは 543648 バイトです
```


②UMB へのロード

AUTOEXEC.BAT に次の 1 行を追加します。

```
UMBLOAD /UA:4 A:¥DOS¥DOSKEY.COM
```

■ブロック選択のメリット

MS-DOS の DEVICEHIGH/LOADHIGH コマンドを使うと、もっとも大きな UMB ブロックから順に使われます（93 ページの UMB の 3 法則を参照）。UMBLOAD はロードする UMB ブロックを選ぶことができるので、UMB を戦略的に活用できます。

ちなみに、この操作を実行した後の UMB は次のようになります。

```
A:¥>UMBLOAD
UMB LOADER Ver 2.07  Copyright (C) 1994 by MELCO,Inc.Japan

[書式]   d>UMBLOAD </UA:n> filename.COM or EXE
[ｵﾌﾟｼｮﾝ] /UA:n  常駐領域を指定出来ます （省略可能）
               +-> n=使用可能なUMB領域の番号

 No    UMB Area    Size        Status
---- ----------- ------ ----------------
  1  D002 - D7FE  32704 使用可能
  2  DE02 - DFFE   8128 使用可能
  3  E802 - E8DC   3488 使用可能
  4  E8E2 - EA26   5184 Used [EXDISK  ]
  4  EA27 - EB2A   4144 Used [KKCFUNC ]
  4  EB2B - EF00  15696 Used [ATOK    ]
  4  EF01 - EFDC   3504 Used [ATOK    ]
  4  EFDD - EFED    256 使用可能
  4  EFEE - F179   6320 Used [DOSKEY  ]  ←―― ロードした DOSKEY
  4  F17A - F400  10336 使用可能
使用可能なメインメモリのサイズは 543648 バイトです
```

便利な機能を使う

ここでは MELWARE for Win の持つ便利な機能について説明します。

メモリチップのハードエラーを検査する

パソコンに装着されているプロテクトメモリに、ハードウェア的な障害がないかをチェックしましょう。

操　作

①CONFIG.SYS の修正

- エディタで CONFIG.SYS を読み込みます。
- EMS/XMS/VCPI/DPMI など、プロテクトメモリを使うデバイスドライバの行頭に REM を挿入し、CONFIG.SYS を保存します。
- リセットボタンを押し、パソコンを再起動します。

②チェックの実行

- 「MEMCHK ⏎」と入力します。
- ⏎ キーを押します。

※チェック終了後に何かキーを押すと、自動的にリセットがかかります

Windows 上でキャッシュサイズを変更する

現在、ディスクキャッシュ（HYPERDSK.EXE）には 3MB が設定してあります。これを起動中の Windows3.1 上から、一時的に 2MB に変更しましょう。

キャッシュサイズを 2MB に減らす

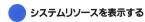
操　作

①デバイスドライバモニタの起動

・「MELWARE for WINDOWS」グループから「デバイスドライバモニタ」をダブルクリックします。

②容量の変更

・［HYPERDSK モニタ（H）］ボタンを左クリックします。
・「変更サイズ」が「2048K バイト」と表示されるまで、← ボタンを左クリックします（→ ボタンを押すと、容量が増える）。
・［終了（C）］→［終了（C）］を左クリックします。

なお、RAM ディスクのサイズを変更するには、CONFIG.SYS の EXDISK.EXE に/W オプションを付ける必要があります。 p.211

システムリソースを表示する

現在、Windows3.1が起動中です。このときのシステムリソース使用率を表示しましょう。

☆システムリソースを表示する

操　作

①リソースモニタの起動

「MELWARE for WiINDOWS」グループから「リソースモニタ」をダブルクリックします。

解　説

■リソースとは

Windows3.1では次の2つのリソースがあります。

リソース名	役　割
GDIリソース	グラフィック処理の情報を管理する64KBのメモリ
USERリソース	Windows上のプログラムを管理する64KBのメモリ

これらは、アプリケーションを複数起動したり、ウィンドウ(ダイアログボックス)を複数開いたりすると、容量が不足して「システムリソースが足りません」などのエラーが表示されます。メモリとリソースの使用率を把握しておくことで、Windows管理を効率的に行えます。

第４章

主要ソフトの快適環境を設定する

ここでは、主なアプリケーションを安全・快適に動かすための、CONFIG.SYSやメモリ環境の設定方法について解説します

Windows3.1 の環境設定

Windows3.1 を快適に動かすための、環境設定について解説します。

CONFIG.SYS を設定する

Windows3.1 に付属するメモリマネージャ「EMM386.EXE」を使用するには、次のような CONFIG.SYS にします。

☆CONFIG.SYS の設定例

```
FILES=35
BUFFERS=20
SHELL=¥COMMAND.COM /P
DEVICE=A:¥WINDOWS¥HIMEM.SYS
DEVICE=A:¥WINDOWS¥EMM386.EXE /P=64 /UMB
DEVICE=A:¥WINDOWS¥SMARTDRV.EXE /DOUBLE_BUFFER
DOS=UMB,HIGH
```

☆AUTOEXEC.BAT の設定例

```
@ECHO OFF
A:¥WINDOWS¥SMARTDRV.EXE
PATH A:¥;A:¥DOS;A:¥WINDOWS
PROMPT $p$g
DOSKEY /INSERT
```

■CONFIG.SYS の SMARTDRV.EXE の必要性

CONFIG.SYS に登録してある SMARTDRV.EXE は、動作速度の遅い初期の SCSI ハードディスクのためのディスクキャッシュ（ダブルバッファマネージャ）です。したがって最新のハードディスクをお使いの場合は、この 1 行は削除した方がメモリの節約になり、また動作スピードも速くなります。ダブル

バッファマネージャが必要かどうかは、MS-DOS のコマンドラインから次のコマンドを実行するとわかります。

```
A:¥WINDOWS¥SMARTDRV ↵
```

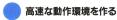
このとき「バッファリング」の列がすべて「不要」と表示される場合は、SMARTDRV.DRV を CONFIG.SYS から削除してもかまいません。なお、AUTOEXEC.BAT に登録されている SMARTDRV.EXE は削除しないでください。

```
A:¥>SMARTDRV
Microsoft SMARTDriveディスク キャッシュ バージョン 4.0
Copyright 1991,1993 Microsoft Corp./ NEC Corporation

キャッシュ サイズ: 2,097,152 バイト
Windows実行時のキャッシュ サイズ: 2,097,152 バイト

          ディスク キャッシュ情報
ドライブ    リードキャッシュ   ライトキャッシュ   バッファリング
------------------------------------------------
  A:        する        する        不要
  B:        する        する        不要
  D:        する        しない      不要
  E:        する        しない      不要

ヘルプを見るには"smartdrv /?"と入力してください。

SMARTDriveのメモリ常駐部分がロードされました。

A:¥>
```

■MS-DOS アプリと共通に使える CONFIG.SYS

p.169

Windows3.1 に付属する EMM386.EXE は VCPI 規格に対応していないため、一太郎 Ver.5 や三四郎などの JW2 対応製品を起動できません。逆に JW2 対応製品に付属する EMS386.SYS は XMS 規格に対応していないため、Windows3.1 を起動できません。双方を快適に動かすには MemoryServer II か MELWARE for Win を使います。それぞれの設定は ***p.169*** を参照してください。

高速な動作環境を作る

Windowsを高速に動かすには、次の4つがポイントになります。

■十分なプロテクトメモリを搭載する

Windowsは複数のアプリケーションを同時に動かすことができます。このときWindowsプログラム・アプリケーションプログラムは、プロテクトメモリ上にロードされます。またクリップデータなどの一時的な作業データも、プロテクトメモリ上に記憶されます。

このときプロテクトメモリに十分な容量が確保できないと、Windowsはハードディスク上の特定領域（スワップファイル）に、プロテクトメモリの未使用データを移動します。ハードディスク上のスワップファイルへのアクセスは、プロテクトメモリへのアクセスに比べて、読み書きの動作が遅くなります。

したがって、Windows上のアプリケーションを高速に動かすには、いかにプロテクトメモリの容量を大きくするかが重要なポイントになります。一般的には、3~4本のアプリケーションを同時に動かすには、8MB以上のプロテクトメモリが必要です。

■高速なCPUを使う

p.10

CPU（パソコンの各種処理を中央で制御する中央演算装置）の処理スピードが速いほど、パソコンの各種動作も速くなります。特にWindowsではフォントの展開・グラフィックデータの処理などCPUに負荷のかかる処理が多いため、低速なCPUを使うと目に見えて画面の動きが重くなります。

インテル製のCPUはi80286・386・486というように、型番の数字が上に行くほど高速になります。また同じ型番でもクロック周波数が大きいほど処理速度は速くなります。たとえばi80486（33MHz）よりi80486（66MHz）の方が2倍のスピードが出ます。Windows3.1を仕事で快適に使いこなすには、実際にはi80486以上は必要です。

最近では倍速CPUチップを装着するだけで、既存のパソコンのCPU能力をi80486相当にグレードアップできる製品が販売されています。倍速CPUチップは使用する機種によって、それぞれ専用のものを使用しなければなりません。購入の際は、メーカーにあらかじめ確認した上で購入してください。

■高速なハードディスクを使う

Windows はハードディスクとプロテクトメモリとの間で、大量のデータをひんぱんに読み書きします。したがって、ハードディスクはデータの読み出し/書き込みが高速な製品の方が、処理スピードも速くなります。一部の初期のSCSI ハードディスクではデータ転送速度が遅いため、Windows を起動するのに 2~3 分もかかるものもあります。これではせっかくの高速な CPU もプロテクトメモリも活かせません。

ハードディスクを選択する際には、パソコンとの接続規格（インターフェイス規格）とデータ転送方式の 2 点に注意してください。

インターフェイス規格

インターフェイス規格には次の 3 種類があります。

規格名	読み方	速度	特　徴
SASI	サシー	×	最大で 40MB までしか管理できない初期の接続規格
SCSI	スカジー	◎	大容量を扱うことができる接続規格
IDE	アイディーイー	○	SASI と SCSI の長所を取り合わせた接続規格

98MATE や 98FELLOW を使っている場合は、IDE 規格のハードディスクがもっとも価格性能比がよいでしょう。複数台のハードディスクを増設する可能性がある場合は、SCSI 規格を選択するとよいでしょう。最近では SCSI II 規格（SCSI 規格の上位互換規格で、さらに高速なアクセスが可能になる）に準拠した製品もあります。

データ転送方式

パソコンとハードディスクとのデータ送受信の手順・方式です。代表的なデータ転送方式には次の 3 つがあります。

転送方式	読み方	速度	特　徴
DMA	ディエムエー	×	パソコンの DMA チャネルを経由してデータを送る方式
I/O	アイオー	○	CPU にデータ送信を制御させる方式
バスマスタ	バスマスタ	◎	ボード側でデータ送信を制御する方式

■常設スワップファイルを設定する

Windowsをインストールした状態では、スワップファイル（ハードディスクをプロテクトメモリの代わりに使う作業領域）は「一時スワップファイル」になっています。一時スワップファイルはWindowsの起動・終了時に自動作成・削除されるため、ファイルが連続した領域に確保されずに、結果的にアクセススピードが落ちる原因になります。そこで次の手順で常設スワップファイルを作成してください。

①スワップファイルを作成するドライブの、ディスク適正化を実行
②Windowsを起動し、メイングループからコントロールパネルを起動
③［エンハンスドモード］➡［スワップファイルの設定（V）］➡［変更（C）］を選択
④［種類（T）］には「常設」を選択
⑤［OK］➡［はい（Y）］を左クリック

なお、このとき［BIOSを経由しないでスワップファイルを利用（U）］を左クリックし、チェックボックスをON（×印が表示された状態）にしておくと、スワップファイルとのアクセスをMS-DOSを経由しないで高速に実行できます。ただし、使用する環境によってはこのチェックボックスを選択できない場合もあります。

常設スワップファイルを作成する

一太郎 Ver.5 の環境設定

JW2 対応製品を快適に動かすための、環境設定について解説します。

製品に付属するメモリマネージャを使う

一太郎 Ver.5 など JW2 対応製品に付属するメモリマネージャ「EMS386.SYS」を使用するには、次のような CONFIG.SYS にします。

CONFIG.SYS 設定例

```
BUFFERS=20
FILES=30
FCBS=1
DEVICE=A:¥ATOK8¥EMS386.SYS /X
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
SHELL=A:¥COMMAND.COM /P
DOS=HIGH
```

AUTOEXEC.BAT 設定例

```
@ECHO OFF
PATH A:¥;A:¥DOS;A:¥JUST5
SET JW2P=A:¥GORO;A:¥HANA3;A:¥SANSIRO;A:¥TARO5;
A:¥JW2;A:¥JEDIT
PROMPT $p$g
DOSKEY /INSERT
```

解　説

■サウンド機能の OFF

p.226

98 シリーズは FM 音源などのサウンド機能を内蔵し、このサウンド機能のプログラム(サウンド BIOS)は、C0000h 以降に割り振られています。EMM386.SYS は、EMS ページフレームを C0000h~CFFFFh (64KB) に確保しようとするためにサウンド BIOS とぶつかってしまい、EMS メモリが使えなくなります。したがって、この場合はパソコンのサウンド機能を OFF にしてください。

MS-DOSのメモリマネージャを使う

JW2はVCPI規格に対応したアプリケーションです。ところがNEC版MS-DOS Ver.5.0はVCPI規格に対応していません。この環境でJW2を起動するには、次のようなCONFIG.SYSにします。

✩MS-DOS Ver.5.0を使う場合

```
BUFFERS=20
FILES=30
FCBS=1
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
SHELL=A:¥¥COMMAND.COM /P
```

✩MS-DOS Ver.5.0Aを使う場合

```
BUFFERS=20
FILES=30
FCBS=1
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=128 /UMB
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
SHELL=A:¥COMMAND.COM /P
DOS=HIGH,UMB
```

なお、MS-DOS Ver.5.0Aでは AUTOEXEC.BATの中に「DPMI」を追加してください。これは、DPMI規格がJW2が対応するVCPI規格の、上位互換の関係にあるためです。ただし、DPMIコマンドを実行すると動作は遅くなります。

Windowsと共通のCONFIG.SYSを使う

JW2 対応アプリケーションと Windows3.1 を共通の CONFIG.SYS で使用するには、MemoryServer II か MELWARE for Win を使います。

◇MemoryServer IIを使う場合

```
BUFFERS=20
FILES=30
FCBS=1
DEVICE=A:¥MDEV¥IOSPRO¥VMM386.EXE /I /U /M /NECID
DEVICE=A:¥MDEV¥IOSPRO¥DC10.EXE 3072 /W=2048 /S
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
SHELL=A:¥COMMAND.COM /P
DOS=HIGH, UMB
```

なお、旧バージョン (MemoryServer) をお使いの場合、VMM386.SYS に/ND (DPMI サーバを起動しない) を付けてください。MemoryServer IIでは DPMI32 コマンドを実行しても問題はありません。

◇MELWARE for Winを使う場合

```
BUFFERS=20
FILES=30
FCBS=1
DEVICE=A:¥MEL4WIN¥MELEMM.386 /HM /M D0, D4, D8, DC /SW1
DEVICE=A:¥MEL4WIN¥HYPERDSK.EXE C:3072 CW:3072
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
SHELL=A:¥COMMAND.COM /P
DOS=HIGH, UMB
```

古いバージョン (Ver.5.17~Ver.5.19) の MELEMM.386 をお使いの場合、ハードディスクなどの SCSI ボードの割り込みレベル (INT) を 1 または 6 に設定すると、JW2 を起動できなくなることがあります。

JW2のメモリ環境を設定する

JW2対応アプリケーション（一太郎Ver.5/三四郎/花子/五郎など）を快適に動かすには、JW2でのメモリ設定がポイントになります。ここではプロテクトメモリが8MBほどあると仮定して、実際に設定してみましょう。

☆JW2のメモリ環境を設定する

操作

①ジャストウィンドウの起動

MS-DOSのコマンドプロンプトから、「JW⏎」と入力します。

②環境設定ユーティリティの起動

- ESC ➡［G・画面］➡［P・プログラム一覧］を選択します。
- ［JW環境設定］➡［K・環境設定］を選択します。

③メモリサイズの設定

- 「<メモリ>サイズ」には、「3072」と入力します。
- 「<ファイル>使用」には、［する］を選択します（ファイルを使用する）。
- 「サイズ」には「4096」と入力し、⏎キーを押します。

④ジャストウィンドウの終了

- ［Q・終了］を選択し、「JW環境設定」を終了します。
- ESC ➡［Q・終了］➡［Y・はい］を選択し、JW2を終了します。

解　説

■[K・環境設定]の各項目の設定

[K・環境設定]の各項目には次のような機能があります。

共通作業ディレクトリ

プロテクトメモリに作業領域（一時的に使う記憶領域）を確保するスペースがないときに、どのディレクトリに作業領域を確保するかを指定します。

自動バックアップディレクトリ

バックアップファイルを作成するディレクトリを、ハードディスク（または不揮発性 RAM ディスク）に指定します。バックアップファイルとは突発的なハングアップなどに備えて、編集内容を保存するファイルのことです。

編集作業領域

文書データや図形データを編集するための領域を指定します。編集できるデータ量を超えると、「編集作業領域の上限に達しました」と表示されます。この場合ここの数値を増やすと、さらに大きなデータを扱えるようになります。

三四郎や一太郎を高速に動作させるには、「メモリ」の領域をできるだけ大きく設定することがコツです。

ただし、プロテクトメモリはプログラムの動作用にも使われるので、「メモリ」の値を大きくしすぎると、逆にプログラムの動作スピードが低下することがあります。プロテクトメモリに 2MB~4MB が残るように「メモリ」を設定することが重要です。

スワップファイル

スワップファイルの大きさを 16KB 単位で指定します。スワップファイルとはプログラムが使用できるプロテクトメモリが足りなくなったときに、動作していないプログラムをハードディスク（または RAM ディスク）に一時的に退避させるファイルです。したがって、スワップファイルが動作する状態になると操作スピードは著しく低下します。

フォントキャッシュを設定する

一太郎 Ver.5 の印刷やイメージ編集モードでの表示では、ハードディスクのアウトラインフォントを1文字ずつ読み込みながら処理します。フォントキャッシュを大きくするとハードディスクのフォントファイルへのアクセス回数を減らすことができます。フォントキャッシュを2048KB (2MB) に設定してみましょう。

☆フォントキャッシュの容量を大きくする

操 作

①ジャストウィンドウの起動

MS-DOS のコマンドプロンプトから、「JW⏎」と入力します。

②フォント設定ユーティリティの起動

・ESC ➡ [G・画面] ➡ [P・プログラム一覧] を選択します。

・[JW 環境設定] ➡ [F・フォント設定] ➡ [C・キャッシュ設定] を選択します。

③キャッシュ領域の指定

・「使用」には「する」を選択します。

・「サイズ」には「2048⏎」と入力します。

・「最大文字サイズ」には「256⏎」と入力します。

これで次回起動時から、フォントキャッシュが設定した容量にセットされます。

解　説

■[F・フォント設定]の各項目の設定

[F・フォント設定]の各項目には次のような機能があります。

キャッシュ領域

フォントキャッシュの容量を64KB単位で指定します。「フォントキャッシュ」とは一度読み込んだフォント情報をメモリ内に溜めておく領域を言います。この領域が大きいほど、ハードディスクへのアクセス回数は減り、結果的に動作速度は速くなります。2048~3072KBを設定すると、イメージ編集モードのスクロールがスムーズになります。しかし、搭載しているプロテクトメモリの容量が少ない場合は、設定する容量に注意してください。

最大文字サイズ

キャッシュ対象とする文字サイズの上限を1ドット単位で指定します。例外的な大きな文字サイズをキャッシュの対象からはずすことで、キャッシュ容量を有効に使うことができます。ドット数は次の式で求めます。

ドット数 ＝ ポイント ÷ 72 × プリンタのDPI値

■さらに高速に動かすには

p.161

JW2をさらに高速に動かすには、ディスクキャッシュを2MB~3MBほどに設定するとよいでしょう。ソフトウェア的に対応できるのはここまでです。これ以上高速にするにはプロテクトメモリを増設したり、CPUをアップグレードするしかありません。

一太郎Ver.5 for Windowsの環境設定

一太郎 Ver.5 for Windows (以下「一太郎 Win」と記述) を快適に動かすための、環境設定について解説します。

編集作業領域のサイズを設定する

一太郎 Win で大きな文書を同時に複数編集しようとしたところ、「編集作業領域の上限に達したため読み込めません」というメッセージが表示されました。そこでこの「編集作業領域」のメモリサイズを 2048KB (2MB) に、ファイルサイズを 4096KB (4MB) に拡大しましょう。

編集領域のサイズを設定する

操 作

①JS 環境設定ユーティリティの起動

[ファイル (F)] ➡ [システム設定 (E)] ➡ [環境設定 (E)] ➡ [編集作業領域] を選択します。

②〈メモリ〉の設定

[サイズ (M)] には「2048」と入力します (2048KB)。

③〈ファイル〉の設定

・[使用 (F)] の [する] を左クリックします。

・［サイズ（S）］には「4096」と入力します（4096KB=約4MB）。
・［OK］ボタンを左クリックします。

④Windows3.1の再起動

［再起動］ボタンを左クリックします。

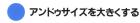
■編集作業領域とは

編集作業領域とは、一太郎Winが文書を編集するのに一時的に使用する作業領域のことです。編集作業領域は次の4つの項目を設定します。

項　目	機　能
〈メモリ〉サイズ（M）	プロテクトメモリ上に確保する領域を64KB単位で入力
〈ファイル〉使用（F）	メモリ上の編集作業領域が上限に達したときに、ハードディスク上に編集作業領域を確保するかどうかを指定
サイズ（S）	［使用（F）］を「する」に設定した場合にハードディスク上に確保する編集領域の容量を16KB単位で入力
ディレクトリ（D）	ハードディスク上に編集領域を確保する場合、どのディレクトリに確保するかを指定（既定値はA:¥JSKIB¥）

■メモリの設定容量に注意

一太郎Winは起動時に「〈メモリ〉サイズ（M）」に設定した容量の作業領域を、装着しているプロテクトメモリから強制的に確保します。他のアプリケーションがメモリを要求しても、占有したまま解放しません。したがって「〈メモリ〉サイズ（M）」の値を大きく取りすぎると、他のアプリケーションに影響が出ます。むしろ「〈メモリ〉サイズ（M）」は小さめに設定し、「〈ファイル〉サイズ（S）」の値を大きめに設定しておいた方が合理的でしょう。次期バージョンでは、状況に応じて一太郎Winがメモリの取得・解放を自動的に行うようにしてほしいものです。なお、使用できるプロテクトメモリ以上の「〈メモリ〉サイズ（M）」を設定すると、起動時に一太郎Winが自動的に値を調整します。また、「ディレクトリ（D）」にはフロッピーディスク・光磁気ディスクなどの交換可能なドライブは指定できません。ハードディスクかRAMディスクのみが指定できます。

アンドゥサイズを大きくする

アンドゥ機能で過去にさかのぼって、次々に編集内容を復元したいと思います。このとき、できるかぎり過去にさかのぼれるようにするために、アンドゥ領域のサイズを、設定できる最大の値を入力しましょう。

☆アンドゥ領域のサイズを設定する

操 作

① [ファイル (F)] コマンドの実行

［ファイル (F) ］➡［システム設定 (E) ］➡［ファイル (F) ］を選択します。

②アンドゥサイズの設定

・［アンドゥサイズ (U) ］には「2048」と入力します (2048KB=約2MB)。
・［OK］ボタンを左クリックします。

※入力できる容量は設定してある編集作業領域によって異なります

解　説

■アンドゥサイズとは

アンドゥ機能は、編集作業領域上にアンドゥファイル（作業の1つ1つを記録する一時作業ファイル）を作成します。CTRL＋Z キーでアンドウ機能を実行すると、このアンドゥファイルを見て、次々と過去の編集状態に戻る仕組みになっています。アンドゥサイズは、このファイルの大きさを設定します。アンドゥファイルの上限まで記録されると、古い操作履歴は削除されます。アンドゥサイズが大きいほど、操作履歴を多く記録できるわけです。

なお、複数回のアンドゥ機能を利用できるようにするには、［ファイル(F)］➡［システム設定(E)］➡［システム(S)］を実行し、［アンドゥを1回に制限(X)］をOFFに設定する必要があります。

■設定できるアンドゥサイズ

アンドゥサイズは、編集作業領域（［サイズ(M)］［サイズ(S)］の合計値）で設定してある値の最大25％までしか設定できません。アンドゥサイズをさらに大きくするには、編集作業領域のサイズを大きくしてください。

Lotus1-2-3R2.4Jの環境設定

Lotus1-2-3R2.4Jを快適に動かすための、環境設定について解説します。

メモリ環境を設定する

現在、EMSメモリを管理するメモリマネージャが、CONFIG.SYSに登録してあります。Lotus1-2-3R2.4Jでできるだけ大きな表を扱えるようにするために、このEMSメモリを使いこなす環境を設定しましょう。

操　作

①環境設定プログラムの起動

・Lotus1-2-3R2.4Jのメニューを表示します(LOTUS.COM)。

・［I・環境設定］を選択します。

☆環境設定プログラムのメニュー

②EMSの設定

・［現在のドライバセットの変更］を選択します。

・「EMS」の項目までカーソルを移動し、スペースキーを押します。
・「拡張メモリのみ」にカーソルを移動し、スペースキーを押します（選択した項目には√マークが付く）。
・⏎キーを押します。

③設定内容の登録

・「ドライバセット名」までカーソルを移動し、⏎キーを押します。
・「これでよろしいですか？」と表示されるので、「はい」を選択します。
・「保存しますか」と表示されるので、「はい」を選択します。
・⏎キーを２回押し、［終了］➡［E 終了］を選択します。

解　説

■3つのメモリ使用方法

環境設定プログラムでは、メモリの使用方法を次の３つから選択できます。

拡張メモリのみ

ワークシートのデータをすべてEMSメモリ上に置きます。もっとも大きな容量（最大5MBまで）を確保できます。

内部メモリと拡張メモリ（規定値）

コンベンショナルメモリにセルポインタ情報を、EMSメモリにセルデータを置きます。約1MBまでのワークシートを扱えます。

使用しません

コンベンショナルメモリのみにワークシートを置きます。小さなワークシートしか扱えなくなります。

リファレンス編

最適なCONFIG.SYSを作成するために必要な
CONFIGコマンド・デバイスドライバの使い方、
また、CONFIG設定時のトラブル対策を
リファレンス的に解説します

リファレンスの構成

イーエムエスディスク・シス　一太郎 Ver.5

EMSDISK.SYS

●メモリ上にRAMディスクを設定する

EMS386.SYSで確保したEMSメモリ上に指定した容量のRAMディスクを設定します。

書　式

```
DEVICE=[d:][path]EMSDISK.SYS /K=nnnnnK
```

パラメータ	設定内容
/K	RAMディスクとして確保するメモリ容量をKB単位で指定する

使用例

●RAMディスクに2MB (2048KB) を割り当てる

```
DEVICE=EMSDISK.SYS /K=2048K
```

解　説

▼RAMディスクとは

RAMディスクとはメモリ上に作るドライブのことで、ハードディスクなどに比べて高速なデータの読み書きが可能です。98シリーズでは、RAMディスクは接続されているドライブの最終ドライブ名が割り振られます。

EMSDISK.SYSは必ずEMS386.SYSより後に記述してください。また他のメーカーのメモリマネージャとは共存できません。

書式の記号の意味

記　号	意　　味
[]	省略可能なオプション
d:	ドライブ名（たとえばBドライブは「B:」とする）
path	パス名（たとえば「¥DOS」とする）
...	以降のオプションを繰り返して指定可能
n	任意の数値（nnなら2桁の数値を表す）
x｜y	xまたはyを指定する

隠しオプションの使用について

リファレンス編ではコマンドやデバイスドライバの隠しオプション（製品マニュアルに記載されていない非公開オプション）についても解説しています。ただしこれらのオプションは、その性格上、メーカーの正式サポートの対象になりません。

したがって、ご使用の際には自己責任のもと十分な注意を払って行ってください。

第1章

CONFIG コマンド一覧

CONFIG.SYS（コンフィグ・シス）とは、MS-DOSの動作環境を設定するためのファイルです
ここではNEC版のMS-DOS Ver.5.0Aが提供する、11個のCONFIG.SYSコマンドを解説します

ブレーク

BREAK

●プログラム中止命令の実行を制御する

プログラム実行中にブレーク信号を受信したとき、どのレベルまで中止命令をきかせるかを設定します。

書 式

```
BREAK=[ON | OFF]
```

パラメータ	設定内容
ON	すべてのDOS機能実行時にブレーク信号をチェックする
OFF	画面・プリンタの入出力時にブレーク信号をチェックする (既定値)

使用例

● BREAK設定をONにする

```
BREAK=ON
```

解 説

▼ブレーク信号とは

ブレーク信号とはMS-DOSの機能(ファンクションリクエスト)を実行しているときに、パソコンから送る中断命令のことを言います。98シリーズではブレーク信号は CTRL + C キーまたは STOP キーに割り振られています。

「BREAK ON」が指定されると、CPUは常にブレーク信号を監視します。その分、いくらかはCPUに負荷がかかる理屈になります。特別な目的がない場合は、OFF(既定値)のままにしておいた方が賢明です。

▼中断できないケース

ブレーク信号でプログラムの中止を実行できるのは、MS-DOSが提供する機能(ファンクション)を利用しているときに限ります。またアプリケーションのほとんどはブレークキー(CTRL + C キー)を無効に設定しています。したがってBREAKコマンドはMS-DOSコマンド実行時の中止を制御できる程度の効果しかありません。

バッファーズ

BUFFERS

●ディスクバッファの容量を設定する

ディスクをアクセスする際の、データの一時的な格納エリア「ディスクバッファ」の容量を設定します。これによりディスクのアクセス回数を減らすことができ、結果的に高速なデータ入出力を実現できます。

書　式

```
BUFFERS=n[,m]
```

パラメータ	設定内容
n	ディスクバッファ数を 1~99 の範囲で指定する (既定値は 20)
m	内部バッファ数を 1~8 の範囲で指定する (既定値は 0)

使用例

●ディスクバッファを 26 に設定する

```
BUFFERS=26
```

●内部バッファを 4 に設定する

```
BUFFERS=26,4
```

解　説

▼BUFFERS の仕組み

MS-DOS ではディスクの読み書きの際には必ず「入出力バッファ」と呼ばれる領域に一時的にデータを溜めます。この入出力バッファが満杯になるか、またはデータが終了したところで、まとめて CPU に送られて処理されます。

入出力バッファが小さいとディスクアクセスの頻度が高くなるためにアクセスピードが遅くなり、逆に大きいとディスクアクセスの回数が減る分だけ、速いスピードを実現できます。

▼BUFFERS の既定値

BUFFERS の既定値はパソコンのコンベンショナルメモリ容量によって異なります。最近のパソコンでは、コンベンショナルメモリは 640KB が標準実装されています。

コンベンショナルメモリ容量	既定値	コンベンショナルメモリ容量	既定値
128	2	512	10
256	2	640	20
384	5	768	20

▼BUFFERS が消費するメモリ

「BUFFERS=1」当たりのメモリ消費は、パソコンに接続されているディスクのうち、最大容量のディスクのセクタ長に 20 バイトをプラスした容量になると考えてよいでしょう（完全に同じではありません）。

ディスク容量	セクタ長	BUFFERS=1 当たりの消費量
1MB~64MB	1024B	約 1044B
65MB~128MB	2048B	約 2068B
129MB~2047MB	512 または 256B	約 1044B

BUFFERS で指定したディスクバッファはコンベンショナルメモリ上に作成されます。したがって BUFFERS の値を多く取りすぎると、コンベンショナルメモリの空き容量が小さくなります。むしろ BUFFERS は 20~30 に設定しておき、ディスクキャッシュ（SMARTDRV.SYS など）を 2MB~3MB に設定する方が、ディスクアクセスの高速化という点では効果的です。

▼内部バッファとは

内部バッファ（m）とはディスクバッファ（n）の補助タンクのような役割を持ちます。この数値を指定しておくとディスクバッファ（n）が確保するデータに加え、その先の連続したセクタのデータをも先読みします。つまりディスクキャッシュ（SMARTDRV.SYS）のミニチュア版といった性格です。ディスクアクセスを高速化する手段として便利です。

SMARTDRV.SYS *p.219*

デバイス

DEVICE

●デバイスドライバをメモリに組み込む

デバイスドライバをMS-DOSに組み込みます。通常はコンベンショナルメモリ上にプログラムが常駐します。

書 式

DEVICE=[d:][path]ファイル名 [オプション]

パラメータ	設定内容
d:path	デバイスドライバのあるパスを指定する
ファイル名	デバイスドライバのファイル名を指定する
オプション	デバイスドライバ固有のオプションを指定する

使用例

●プリンタ制御ドライバを組み込む

```
DEVICE=A:¥DOS¥PRINT.SYS /U
```

●メモリ設定ドライバを組み込む

```
DEVICE=A:¥DOS¥HIMEM.SYS
DEVICE=A:¥DOS¥EMM386.EXE /P=128  /UMB
```

解 説

▼デバイスドライバとは

デバイスドライバとは、デバイス（周辺機器）とパソコンとのデータのやりとりを制御するプログラムです。ハードウェアにもっとも近いレベルでデバイスを制御します。

▼デバイスドライバの取り外し方法

DEVICEコマンドで組み込むと、MS-DOSが起動中はデバイスドライバを取り外すことができません。取り外すにはCONFIG.SYSを修正後、リセットボタンを押してパソコンを再起動してください。

デバイスドライバを自由に組み込んだり取り外したりするには、ADDDRV/DELDRV コマンドを利用します。ただし ADDDRV/DELDRV コマンドはキャラクタ系デバイスドライバのみに使用できます。デバイスドライバにはブロック系デバイスドライバ（データをあるまとまった単位で処理するデバイスドライバ）と、キャラクタ系デバイスドライバ（1文字単位で処理を行うデバイスドライバ）があります。

参照 *p.105*

▼デバイスドライバの消費メモリ

デバイスドライバは1つ組み込むごとに、それぞれ必要なメモリを消費します。したがって不必要なデバイスドライバを CONFIG.SYS に残しておくと、メモリの浪費になります。下に示す表は標準的に組み込んだ場合の、コンベンショナルメモリの消費量です。

MS-DOS Ver.5.0A のデバイスドライバのメモリ消費量

デバイスドライバ	容　量	デバイスドライバ	容　量
HIMEM.SYS	3808	FONT.SYS	9664
EMM386.EXE	7808	PRINT.SYS	5312
NECAIK1.DRV	63488	MOUSE.SYS	3456
NECAIK2.DRV	61088	SETVER.SYS	96
SMARTDRV.SYS	24032	RSDRV.SYS	2416
KKCFUNC.SYS	3952	GRAPH.SYS	33568

※単位（バイト）

▼パス指定について 注意

CONFIG.SYS に「DEVICE=A:¥DOS¥MOUSE.SYS」と記述した場合は、実際に「A:¥DOS」に「MOUSE.SYS」というデバイスドライバがないとエラーになります。DEVICE コマンドのパス指定どおりにデバイスドライバが存在するかどうかを、あらかじめ確認する必要があります。

参照 **DEVICEHIGH** *p.189*

デバイスハイ

DEVICEHIGH

●デバイスドライバを UMB へ組み込む

デバイスドライバを UMB（アッパーメモリブロック）へ組み込みます。

書　式

Ⅰ. **DEVICEHIGH=[d:][path]ファイル名　[オプション]**
Ⅱ. **DEVICEHIGH Size=nnnn　[d:][path]ファイル名　[オプション]**

パラメータ	設 定 内 容
d:path	デバイスドライバのあるパスを指定する
ファイル名	デバイスドライバのファイル名を指定する
オプション	デバイスドライバのオプションを指定する
nnnn	あらかじめ UMB に確保するメモリ容量を 16 進数で指定する

使用例

●プリンタ制御ドライバを UMB に組み込む

```
DEVICEHIGH=A:¥DOS¥PRINT.SYS /U
```

●あらかじめ 24KB のメモリを確保してから組み込む

```
DEVICEHIGH Size=5FF0 A:¥DEV¥TESTA.DRV
```

解　説

▼DEVICEHIGH の特徴　*p.59*

DEVICEHIGH で指定したデバイスドライバは UMB 上に常駐されるので、結果的にコンベンショナルメモリを節約することができます。ただし、指定したデバイスドライバを常駐させるために必要なメモリが UMB にない場合は、自動的にコンベンショナルメモリに常駐されます。このとき UMB に常駐できたかどうかのメッセージは表示されません。それを確認するにはメモリ表示ユーティリティ（VMAP・MEM・MSD など）で調べるより方法はありません。

DEVICE *p.187*　**DOS** *p.190*

ド　ス

DOS

●HMA へ MS-DOS システムの一部をロードする

MS-DOS システムの一部を HMA へロード (配置) したり、UMB の使用許可を設定します。コンベンショナルメモリを広く使うためのコマンドです。

書　式

I. **DOS=[HIGH | LOW,]{UMB | NOUMB}**
II. **DOS={HIGH | LOW, }[UMB | NOUMB]**

パラメータ	設 定 内 容
HIGH	MS-DOS の一部を HMA へロードする
LOW	HMA を使用しない (既定値)
UMB	UMB の使用を許可する
NOUMB	UMB を使用しない (既定値)

使用例

●HMA メモリへ MS-DOS の一部をロードする

```
DOS=HIGH
```

●複合指定する

```
DOS=HIGH, UMB      ←――― HMA/UMB を使用する
DOS=HIGH, NOUMB    ←――― HMA のみを使用する
```

解　説

▼DOS コマンドの用途

HIGH オプションを設定すると起動時に MS-DOS の一部 (カーネル部) が HMA へロードされ、コンベンショナルメモリは約 60KB 広くなります。また UMB オプションは UMB の使用許可を与えるものです。実際にメモリマネージャで UMB を確保していないと、UMB オプションの設定は機能しません。

EMM386.EXE *p.201*　**DEVICEHIGH** *p.189*

エフシービーエス

FCBS

●FCBで同時にオープンできるファイル数を指定する

FCB（ファイルコントロールブロック）方式を使って同時にオープンできるファイル数を指定します。FCBはMS-DOS Ver.2.11以前で使用されていた古い方式のファイルオープン規格です。

書 式

```
FCBS=n
```

パラメータ	設定内容
n	同時にオープンできるファイル数を1~255の範囲で指定（既定値は4）

使用例

●同時に20個のファイルを使用できるように指定する

```
FCBS=20
```

解 説

▼FCBの用途

FCB（File Control Block standard）方式でファイルをオープンするのは、MS-DOS Ver.2.11以前に作られた一部のアプリケーションのみで、現在では対応しているアプリケーションはほとんどありません。通常はファイルハンドル方式（FILESコマンド）を使用します。

▼メモリを有効利用する

FCBSは設定値1当たり約48バイトを消費します。古いアプリケーションとの互換性が問題にならない限り、このFCBSは最低値の1に設定し、メモリを節約するとよいでしょう。

```
FCBS=1
```

FILES *p.192*

ファイルズ

FILES

●同時にオープンするファイル数を設定する

ファイルハンドル方式を使って、同時にオープンできるファイル数の最大値を設定します。

書　式

```
FILES=n
```

パラメータ	設定内容
n	同時にオープンできるファイル数を 8~255 の範囲で指定（既定値は 8）

使用例

●同時に 30 個のファイルをオープンできるようにする

```
FILES=30
```

解　説

▼ファイルハンドル方式とは

MS-DOS は起動時に指定された FILES 値の数だけ、コンベンショナルメモリにファイルテーブルを確保します。アプリケーションはファイルをオープン（ファンクションリクエストの 2FH~60H を使用）する際に、このファイルテーブルを使用してファイル操作を行います。このファイルテーブルを使用するときに MS-DOS が与える番号を「ファイルハンドル」と言います。同じファイルオープンでも、FCBS コマンドはファイルコントロールブロック方式を使っています。現在では FILES コマンドを使用するのが一般的になっています。

▼使用時の注意点　注意

Windows3.1 では、最低でも FILE 値には 30 以上を指定しないと起動できません。ACCESS をインストールすると 100 が自動指定されます。なお、「FILES=1」当たりのメモリ消費量は約 54 バイトです。意味もなく FILES 値を多くするとメモリの浪費につながります。

インストール

INSTALL

●常駐型プログラムをCONFIG.SYSから組み込む

起動時にTSR（常駐型プログラム）をコンベンショナルメモリに組み込みます。起動時に1回だけ実行すればよいようなTSRを指定します。

書 式

```
INSTALL=[d:][path]ファイル名　[オプション]
```

パラメータ	設定内容
d:path	常駐型プログラムのあるパス名を指定する
ファイル名	常駐型プログラムのファイル名を指定する
オプション	プログラム固有のオプションを指定する

使用例

●常駐型プログラムを組み込む

```
INSTALL=A:¥DOS¥MOUSE.COM ←─── MOUSEコマンド
INSTALL=A:¥DOS¥FASTOPEN.EXE A:80 B:100 ←─── FASTOPENコマンド
```

解 説

▼使用時の注意点

INSTALLコマンド実行時には、次の点に注意が必要です。

TSRの解放は不可

INSTALLコマンドで組み込んだTSRは、コマンドラインから解除（常駐したメモリの開放）できません。したがって、不必要なTSRをINSTALLコマンドで組み込むとメモリの浪費になります。

指定できないTSR

組み込むときに環境変数を参照するタイプのTSRは、正常な動作をしない場合があります。これはCONFIG.SYSはAUTOEXEC.BATよりも先に展開されるため、AUTOEXEC.BATで設定する環境変数を参照できないためです。たとえば「ですくきっと」（まつもと）がそれに該当します。

ラストドライブ

LASTDRIVE

●アクセスできる最終ドライブ名を指定する

ネットワークまたはSUBSTコマンドで使用する仮想ドライブの、最終ドライブ名を指定します。

書　式

```
LASETDRIVE=x
```

パラメータ	設 定 内 容
x	最終ドライブ名をA~Zの範囲で指定する（既定値はE）

使用例

●最終ドライブを指定する

```
LASTDRIVE=H  ←―― Hドライブ
LASTDRIVE=G  ←―― Gドライブ
```

解　説

▼LASTDRIVEの特徴

LASTDRIVEコマンドはネットワークまたは仮想ドライブを使用する以外は使用しません。既定値はEドライブになっていますが、実ドライブが「E」より多い場合は、接続されている最終ドライブが最終値になります。また、実際のドライブ構成より小さなドライブ番号をLASTDRIVEコマンドで指定すると、実際のドライブ構成の方が優先されます。

▼消費するメモリ

LASTDRIVE値の1ドライブ当たり約88バイトのコンベンショナルメモリを消費します。したがって仮想ドライブを使用する場合には、必要最低限の値に設定しないとメモリを浪費します。

レ　ム

REM

●CONFIG.SYS 中に注釈文を記述する

CONFIG.SYS の任意の行に、処理への影響がないコメント（注釈文）を記述します。

書　式

REM 注釈文

パラメータ	設定内容
注釈文	任意の文字列を記述する

使用例

●既存の CONFIG.SYS コマンドを無効にする

```
DEVICE=A:¥DOS¥EMM386.EXE /P=244 /UMB      ←── 修正前
                  ↓
REM DEVICE=A:¥DOS¥EMM386.EXE /P=244  /UMB ←── 修正後
```

●設定内容の意味をメモ書きする

```
REM ===== 一太郎 Ver.5 専用の CONFIG.SYS ======
```

解　説

▼REM コマンドの用途

MS-DOS は起動時に CONFIG.SYS の中に REM コマンドを発見すると、その行はコメント行と解釈します。この機能を利用してコメントを書き込むと、CONFIG.SYS は格段に見やすくなります。

また REM コマンドは既存の CONFIG コマンドの先頭に書き込むことによって、その機能を一時的に停止させる使い方があります。コマンドのオプションをいろいろと試行錯誤するときに便利です。REM コマンドの行は 10 行あっても 100 行あっても、メモリはいっさい消費しません。

シェル

SHELL

●コマンドプロセッサを指定する

MS-DOS が使用するコマンドプロセッサ（シェルプログラム）を指定します。

書　式

SHELL=[d:][path]プログラム名　[オプション]

パラメータ	設定内容
プログラム名	コマンドプロセッサ名を指定する（既定値は¥COMMAND.COM /P）
オプション	コマンドプロセッサのオプションを指定する

使用例

●COMMAND.COM の再ロード先を指定する

```
SHELL=A:¥COMMAND.COM A:¥ /P
SHELL=B:¥DOS¥COMMAND.COM B:¥DOS /P
```

●環境変数エリアを 1024 バイトに拡大する

```
SHELL=A:¥COMMAND.COM A:¥ /P /E:1024
```

●起動時に AUTOEXEC.BAT を実行しないようにする

```
SHELL=A:¥COMMAND.COM A:¥ /P /D
```

解　説

▼コマンドプロセッサとは

コマンドプロセッサとはキーボードやアプリケーションからの命令を受け付けて、MS-DOS 本体へ翻訳して伝えるプログラムです。MS-DOS の最初の設計段階では、ユーザーが自在にコマンドプロセッサを取り替えられるように考えていたようです。しかし事実上、コマンドプロセッサは MS-DOS に付属する COMMAND.COM しかありません。

コマンドプロンプトを表示したり、入力されたコマンドを MS-DOS システムに伝え、その結果を画面に表示したりするのは、すべて COMMAND.COM が行っています。

▼COMMAND.COM のオプション

p.21

SHELL コマンド自体には、コマンドプロセッサ名を指定する以外は特別なオプションはありません。/P や/E などは COMMAND.COM のオプションです。COMMAND.COM は次のようなオプションがあります。

書 式

COMMAND [[d:]path][CTTY デバイス名] [/E:環境エリア容量] [/P][/C[path]コマンド][/MSG]

パラメータ	設 定 内 容
d:path	COMMAND.COM の再ロードする場合のパス名を指定する
CTTY	コマンド入出力を行うデバイス名を指定する
/E:	設定する環境変数エリアの容量（バイト単位）で指定する
/P	COMMAND.COM からの抜け出しを禁止する
/C	子プロセスで実行するコマンドを指定する（実行後に復帰）
/MSG	COMMAND のエラーメッセージをすべてメモリに常駐させる
/D	AUTOEXEC.BAT を実行しない

▼環境変数エリア設定のコツ

/E オプションは環境変数エリア（既定値は 256 バイト）の大きさを変更するためのものです。バッチプログラムなどで多くの環境変数を使用する場合のみに使用します。環境変数エリアは必ず 16KB 単位で増減するので、16 の倍数で指定するとメモリを効率的に使用できます。環境変数エリアを大きくしすぎると、コンベンショナルメモリを圧迫するので、必要容量だけを指定してください。

第２章

MS-DOS デバイスドライバ一覧

ここでは、NEC版MS-DOS Ver.5.0Aが提供する、16個のデバイスドライバについて解説します
デバイスドライバとは、DEVICE（またはDEVICEHIGH）コマンドで、CONFIG.SYSに登録するプログラムです

イーエムエム・シス

EMM.SYS

●16ビットマシンでEMSメモリを使用する

ハードウェアEMSボード（主に16ビットCPU搭載のパソコンで使用）を使用するパソコンで、EMSメモリを使用できるようにします。

書式

DEVICE=[d:][path]EMM.SYS [/P=ページ数]
[/F=フレームアドレス][/H=ハンドル数]

パラメータ	設定内容
/P	確保するEMSページ数を指定する（1ページは16KB）
/F	物理ページの開始アドレスを指定する
/H	EMSアクセス用のハンドル数を1~255の範囲で指定する（既定値は64）

使用例

●2048KB（16KB×128ページ）のEMSメモリを確保する

DEVICE=EMM.SYS /P=128

●EMSページフレームをD000に移動する

DEVICE=EMM.SYS /F=D000

解説

▼EMM.SYSの用途

EMM.SYSは主に16ビットマシンの拡張スロットに挿入するEMS専用ボードを、EMSメモリとして使用するために使います。このEMS専用ボードはEMSページの切り替えをボード自身が行うため、「ハードウェアEMS」と呼ばれています。

なおEMM.SYSは、32ビットマシンのプロテクトメモリをEMSメモリとして使用することもできます。しかしこの場合はUMBメモリを確保できないので、せっかくのメモリ資源が有効に利用できません。したがってi80386以上のCPUを積んだパソコンを使用する場合は、EMM386.EXEを使用してください。

参照 EMM386.EXE *p.201* HIMEM.SYS *p.206*

イーエムエム 386・イグゼ

EMM386.EXE

●32 ビットマシンで XMS メモリを使う

HIMEM.SYS からもらい受けたプロテクトメモリを、EMS メモリや UMB メモリとして使用できるようにします。NEC 版は VCPI/DPMI 規格には未対応です。

書 式

DEVICE=[d:][path]EMM386.EXE [/M=メモリサイズ | /P=ページ数]
[/F=フレームスタート[-フレームエンド | [/D] | [/U]]
[/L[=ローエストフレームスタート]][/I=mmmm-nnnn]
[/E=mmmm-nnnn][/UMB][/NOEMS]
[/A=高速代替マップレジスタ数][/H=ハンドル数]
[/T=[d:][path]EXTDSWAP.SYS][/MOVEHDBIOS]

パラメータ	設 定 内 容
/M	確保する EMS 容量を 16KB~32768KB の範囲で指定する
/P	確保する EMS ページ数を 1~2048 の範囲で指定する
/F	物理ページのアドレス範囲を指定する（既定値は/F=C000-DFFF）
/D	物理ページを下半分（2 ページ）だけ確保する
/U	物理ページを上半分（2 ページ）だけ確保する
/L	ラージページ EMS モードを指定する。開始セグメントを指定する場合は 1000~4000 の範囲で指定する（既定値は 4000）
/UMB	UMA の空き領域を UMB に割り当てる
/NOEMS	EMS メモリを使用禁止し、UMB のみを使用する
/A	高速代替マップレジスタを 0~254 の範囲で指定する（既定値は 7）
/H	ハンドル数を 2~254 の範囲で指定する（既定値は 64）
/T	拡張タスクスワップ機能を組み込む（MS-DOS Ver.5.0A 以上）
隠しパラメータ	設 定 内 容
/E	UMB 領域に割り当ててはならない領域を指定する
/I	拡張 ROM エリアに強制的に UMB 領域を割り当てる
/MOVEHDBIOS	HDD の ROM アドレスを移動し UMB を広くする

使用例

●2048KB の EMS メモリを確保する

```
DEVICE=EMM386.EXE /M=2048   ←――― サイズで指定した場合
DEVICE=EMM386.EXE /P=128    ←――― ページで指定した場合
```

●EMS 物理ページの開始アドレスを変更する

```
DEVICE=EMM386.EXE /P=128 /F=B000-BFFF
```

●UMB を使用できるようにする

```
DEVICE=EMM386.EXE /UMB
```

●EMS メモリを不使用にする

```
DEVICE=EMM386.EXE /NOEMS
```

●EMS 上で使用するハンドル数の最大値を 50 に設定する

```
DEVICE=EMM386.EXE /P=64 /H=50
```

●C000-CFFF に UMB に割り当てないようにする

```
DEVICE=EMM386.EXE /UMB /E=C000-CFFF
```

●高速マルチタスク環境のための代替レジスタ数を 10 に指定する

```
DEVICE=EMM386.EXE /A=10
```

※A=1 当たり 200 バイトのメモリを消費

●ROM BASIC 領域 (E800-F4FF) に UMB メモリを強制的に割り当てる

```
DEVICE=EMM386.EXE /UMB /I=E800-F4FF
```

●ハードディスク BIOS を移動し、UMB をできるだけ広くする

```
DEVICE=EMM386.EXE /UMB /MOVEHDBIOS /I=D000-DFFF
```

●拡張タスクスワップ機能を組み込む (MS-DOS Ver.5.0A のみ)

```
DEVICE=EMM386.EXE /P=128 /T=A:¥DOS¥EXTDSWAP.SYS
```

解　説

▼EMM386.EXE の用途

EMM386.EXE は HIMEM.SYS が確保したプロテクトメモリを XMS 規格に準拠して、EMS メモリや UMB メモリに割り振ります。HIMEM.SYS と EMM386.EXE は MS-DOS Ver.5.0 のメモリ管理の中心となるデバイスドライバです。マイクロソフト版では EMM386.EXE は VCPI 規格に対応していますが、残念ながら NEC 版では非対応に改変されています。NEC 版の EMM386.EXE を SYMDEB コマンドで VCPI 規格対応に改造することもできます。しかし必ずしも VCPI 完全対応になるわけではなく、一部では動作が不安定になるなどの症状が報告されているので、お勧めできません。

▼UMB の使用 *p.190*

/UMB を指定すると EMM386.EXE は自動的に UMA から空き領域を探して、UMB として確保します。ただし CONFIG.SYS に「DOS=UMB」と記述しないと、実際に使えるようにはなりません。

▼拡張タスクスワップ機能について

MS-DOS Ver.5.0 付属の DOS シェルではタスクスワップ時に画面が真っ黒になったり、タスクスワップ先での ADDDRV/DELDRV コマンドが使用できないなどの不具合がありました。MS-DOS Ver.5.0A ではこれら諸問題を解決するために、拡張スワップファイル「EXTDSWAP.SYS」を/T オプションで指定できるようになっています。EMS を 1 ページ (16KB) 消費しますが、DOS シェルを使用する場合は必ず/T を使用してください。

▼EMM386.EXE を使用するための環境

EMM386.EXE は i80386 以上の CPU を搭載したマシンのみで使用可能です。また EMS メモリを設定するには最低 272KB (論理ページ 17 ページ分) の EMS メモリが確保できないと使用することができません。

なお、EMM386.EXE は HIMEM.SYS より後の行に記述しなければなりません。また EMM386.EXE を DEVICEHIGH コマンドで登録したり、DEVICEHIGH コマンドを EMM386.EXE より前の行で使用できません。

▼EMS 設定の変更について

アプリケーションの中には、一太郎 Ver.3/花子 Ver.2/Z's STAFF KiD98 Ver.3 などのように EMS ページフレームが C0000~CFFFF にないと正常に動かないものが多くあります。またラージページ EMS モードで動くアプリケーションは現在ではほとんど存在しません。したがって/F、/D、/U、/L などのオプションは、既存のアプリケーションが動くかどうかを十分注意してからお使いください。

なお、/F、/M、/P などの EMS に関するオプションが指定されていても、/NOEMS オプションがあるとそれが優先されます。

 DOS *p.190* **HIMEM.SYS** *p.206* **EMM.SYS** *p.200*

フォント・シス

FONT.SYS

●文字フォントを拡大/縮小してアプリケーションに渡す

パソコン内蔵の漢字ROMまたはマルチフォントROMボード（別売）から取得した文字フォントパターンを拡大/縮小して、アプリケーションに渡します。

書　式

```
DEVICE=[d:][path]FONT.SYS  [/M (mmm, nnn) ][/E]
```

パラメータ	設定内容
/M	ROMから受け取るフォントのドットサイズを最大値を指定する
mmm	X軸方向のドット数（既定値は40）。マルチフォントROMボード実装時は8~400、非実装時は8~40から指定する
nnn	Y軸方向のドット数（既定値は40）。マルチフォントROMボード実装時は8~400、非実装時は8~40から指定する
/E	常駐メモリの一部をEMSメモリへロードする

使用例

- フォントドライバをEMSに登録する

```
DEVICE=FONT.SYS /E
```

- 文字フォントの最大値を縦400、横400ドットに指定する

```
DEVICE=FONT.SYS /M (400,400)
```

解　説

▼FONT.SYSの実用性

FONT.SYSはマルチフォントROMを実装したパソコンで、様々な書体・サイズの文字を表示・出力するためのデバイスドライバです。現在ではFONT.SYSに対応したアプリケーションはほとんどありません。またWindowsが普及している現在となっては、FONT.SYSはほとんど使用価値はありません。なお、FONT.SYSが取得したパターンデータをGVRAM（画像データ表示用のメモリ）に出力するには、グラフィックドライバ(GRAPH.SYS)をCONFIG.SYSに登録しておく必要があります。

p.34、p.205

グラフ・シス

GRAPH.SYS

●グラフィック描画機能を使用できるようにする

アプリケーションへ、円弧や直線などのグラフィック描画機能を提供します。

書　式

I. **DEVICE=[d:][path]GRAPH.SYS**
II. **DEVICE=[d:][path]GRAPH.SYS　/F=GRP_H98.LIB　[/E]**

※IIの書式はPC-H98シリーズでのみ使用可能

パラメータ	設定内容
/F	グラフィックライブラリを指定する(PC-H98シリーズ使用時)
/E	常駐メモリの一部をEMSへロードする(PC-H98シリーズ使用時)

使用例

●98シリーズへ登録する

DEVICE=GRAPH.SYS

●PC-H98シリーズへの登録

DEVICE=GRAPH.SYS　/F=GRP_H98.LIB　/E

解　説

▼GRAPH.SYSの実用性

GRAPH.SYSは円弧・矩形・直線などを描画するためのプログラムの集合体です。アプリケーションがこれを利用することで、自身のプログラムをシンプルに作れます。しかし現在では、GRAPH.SYSに対応したアプリケーションはほとんどありません。

▼グラフィックライブラリの格納場所

98シリーズで使用する場合はグラフィックライブラリ「GRAPH.LIB」、PC-H98シリーズでの場合は「GRP_H98.LIB」が、カレントディレクトリまたはGRAPH.SYSと同じディレクトリに格納されていなければなりません。

ハイメム・シス

HIMEM.SYS

●プロテクトメモリをXMSメモリとして管理する

プロテクトメモリをXMS規格で活用できるように管理します（i80286以上のパソコンでのみ使用可能）。これによりEMM386.EXEが使用できるようになります。

書　式

DEVICE=[d:][path]HIMEM.SYS　[/HMAMIN=mm]
[/NUMHANDLES=nnn]

パラメータ	設 定 内 容
/HMAMIN	HMAの使用領域を0~63KBの範囲で指定する（既定値は0）
/NUMHANDLES	EMBの使用ハンドル数を1~128の範囲で指定する（既定値は32）

使用例

●HIMEM.SYSを登録する

DEVICE=HIMEM.SYS

●40KB以上の使用要求があったときHMAを明け渡す

DEVICE=HIMEM.SYS /HMAMIN=40

解　説

▼HMAの使用許可（/HMAMIN）

HMAは1つのプログラムでしか使用できません。/HMAMINは複数のプログラムがHMAを使用しようとした場合、指定容量以上の使用要求があったプログラムに対して使用許可を与えます。既定値0は最初に使用要求を発したプログラムを無条件に優先する指定です。通常は「DOS=HIGH」でMS-DOSシステムをロードします。

▼ハンドル数の設定（/NUMHANDLES）

ハンドルとは各アプリケーションがEMBを使用する際に、重複使用を避けるために与えられる識別番号です。通常は既定値（32）のままでかまいませんが、この数値を減らすとコンベンショナルメモリを1当たり6バイト節約できます。

参照　EMM386.EXE　*p.201*

ケイケイシーファンク・シス

KKCFUNC.SYS

●MS-DOS Ver.5.0 対応 FEP の動作を支援する

MS-DOS Ver.5.0 対応の FEP（日本語変換プログラム）の各種動作を監視/支援します。複数の FEP の登録または複数のタスクからの FEP 起動ができるようになります。

書　式

```
DEVICE=[d:][path]KKCFUNC.SYS
```

使用例

●ATOK8 の先頭に登録する

```
DEVICEHIGH=A:¥DOS¥KKCFUNC.SYS   ←――― UMB を利用する
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICEHIGH=A:¥ATOK8¥ATOK8EX.SYS
```

解　説

▼KKCFUNC.SYS の役割

p.103

KKCFUNC.SYS は複数の FEP を登録したとき、SELKKC コマンドで目的の FEP を選択できるようにします。またタスクスワップなど複数タスクからの同時アクセスで、辞書ファイルが破壊されるのを保護する機能があります。したがって単一の FEP しか使わない場合でも、KKCFUNC.SYS を登録しておくことをお勧めします。ただし MS-DOS Ver.5.0 非対応 FEP は、KKCFUNC.SYS の管理対象にはなりません。

▼CONFIG.SYS への記述位置

KKCFUNC.SYS は必ず FEP ドライバ記述より先の行に記述する必要があります。また KKCFUNC.SYS は同時に 2 行以上記述してはいけません。

KKCSAV.SYS *p.208*

ケイケイシーサブ・シス

KKCSAV.SYS

●MS-DOS Ver.5.0 非対応 FEP の動作を支援する

MS-DOS Ver.5.0 非対応の FEP (日本語変換プログラム) を安定動作させます。

書　式

```
DEVICE=[d:][path]KKCSAV.SYS
```

解　説

▼KKCSAV.SYS の役割

KKCSAV.SYS は MS-DOS Ver.5.0 に対応していない FEP を、安定動作させるためのデバイスドライバです。KKCSAV.SYS は必ず FEP 行の先頭に記述します。また 2 つ以上の KKCSAV.SYS は記述してはいけません。

主な機能としては MS-DOS Ver.5.0 特有の環境への対処 (INT21 への対処、HMA への MS-DOS システムの参照など) があります。しかし、詳細な機能については公開されていません。確かなことは、KKCSAV.SYS を指定したからといってすべての MS-DOS Ver.5.0 非対応 FEP が MS-DOS Ver.5 に完全対応するというわけではないという点です。複数のタスクから MS-DOS Ver.5.0 非対応 FEP を起動すると、辞書の破壊などの事態もあり得ます。あくまで「その場しのぎ」のためのユーティリティにすぎないことを十分に認識しておく必要があります。

▼MS-DOS Ver.5.0 非対応 FEP と DOSSHELL の相性

MS-DOS Ver.5.0 非対応 FEP は、DOSSHELL とは非常に相性が悪いと言えるでしょう。ATOK6 を使用すると DOSSHELL は確実にハングアップします。また、松茸 Ver.3.0 (MS-DOS Ver.5.0 非対応バージョン) では、タスクスワップ時にメモリエラーの警告が表示される場合があります。このように KKCSAV.SYS の有無にかかわらず、MS-DOS Ver.5.0 非対応 FEP と DOSSHELL は、もともと両立しない場合が多いので注意が必要です。

KKCFUNC.SYS *p.207*

マウス（マウスエイチ 98）・シス

MOUSE.SYS (98用)
MOUSEH98.SYS (PC-H98用)

●マウスを制御する

プログラムからマウスを使用できるようにします。

書式

DEVICE=[d:][path]MOUSE.SYS [/I:ベクタ番号]

パラメータ	設定内容
/I	割込ベクタ番号を 11H、12H、14H、15H から指定する (既定値は 15H)

※ハイレゾリューションモードでは 0EH に固定

使用例

●マウスドライバを組み込む

DEVICE=MOUSE.SYS ← 98 シリーズ

DEVICE=MOUSEH98.SYS ← PC-H98 シリーズ

●割り込みベクタ番号を 11H に変更する

DEVICE=MOUSE.SYS /I:11H

解説

▼MOUSE.SYS の実用性

マウスを使用するアプリケーションの多くは独自のマウス制御ルーチンを内蔵しており、現在では MOUSE.SYS はほとんど使われません。また DOSSHELL 内でマウスを使用するには、MOUSE.SYS ではなく MOUSE.COM (MS-DOS コマンド) の方を使用します。

▼割り込みベクタとは

割り込みベクタ番号とは、周辺機器のデータがメモリに転送されるときの「信号」のようなものです。これが他の周辺機器と重複すると、どちらかが機能しなくなるなどの障害が発生します。このような場合に限り、割り込みベクタを変更します。通常は既定値 (15H) のままで、変更の必要はありません。

エヌイーシーエイアイケイ1(2)・ディーアールブイ

NECAIK1.DRV
NECAIK2.DRV

●AIかな漢字変換を使用できるようにする

NECの提供するFEP(日本語入力プログラム)である「AIかな漢字変換プログラム」を使用するためのデバイスドライバです。

書 式

DEVICE=[d:][path]NECAIK1.DRV [/F=d:path キーファイル名]
[/H][/J][/P=読みがな色 変換文字色|M][/W]
DEVICE=[d:][path]NECAIK2.DRV [/T][/R][d:][path]
[辞書ファイル名]

パラメータ	設定内容
/F	使用するキーファイル名 (既定値はNECAIKEY.DAT)
/H	句読点変換をしないように設定する
/J	漢字コードの入力を連続で行えるように設定する
/P	読みがな/変換文字の色を色番号(1~7)で指定する(既定値は/P=5 4)
/W	単語登録を連続して行えるように設定する
/T	変換時にAI機能を使用しないように設定する
/R	連文節変換モードを設定する(未指定時は逐次変換モードになる)
辞書ファイル名	使用する辞書ファイル (既定値はNECAI.SYS)

使用例

●キーファイル「B:¥JFEP¥NECAIKEY.DAT」を使用する

```
DEVICE=NECAIK1.DRV /F=B:¥JFEP¥NECAIKEY.DAT
DEVICE=NECAIK2.DRV
```

●句読点「。、」での変換をしないようにする

```
DEVICE=NECAIK1.DRV /H
DEVICE=NECAIK2.DRV
```

●漢字コード入力を連続して行えるように設定する

DEVICE=NECAIK1.DRV /J

DEVICE=NECAIK2.DRV

●読みがなを緑に、変換文字を赤に文字色を設定する

DEVICE=NECAIK1.DRV /P=3 1

DEVICE=NECAIK2.DRV

●モノクロディスプレイで使用する

DEVICE=NECAIK1.DRV /P=M

DEVICE=NECAIK2.DRV

●単語登録を連続して入力できるように設定する

DEVICE=NECAIK1.DRV /W

DEVICE=NECAIK2.DRV

●漢字変換のときにAI機能を利用しない

DEVICE=NECAIK1.DRV

DEVICE=NECAIK2.DRV /T

●漢字変換モードを連文節変換モードに設定する

DEVICE=NECAIK1.DRV

DEVICE=NECAIK2.DRV /R

●漢字変換モードを逐次変換モードに設定する

DEVICE=NECAIK1.DRV

DEVICE=NECAIK2.DRV

●辞書ファイル「A:¥DOS¥NECAI.SYS」を指定する

DEVICE=NECAIK1.DRV

DEVICE=NECAIK2.DRV A:¥JFEP¥NECAI.SYS

解　説

▼色番号の指定

/Pオプションは入力時の文字色・変換時の文字色を変更するためのものです。色番号には次の7種類があります。

色	赤	青	緑	シアン	黄	紫	白
番号	1	2	3	4	5	6	7

ただし、モノクロディスプレイ(ノート型パソコンなど)を使用する場合は「/P=M」を指定することで、ノーマル・リバース・下線だけで表示できます。

▼逐次(ちくじ)変換機能とは

「きしゃますますごせいえいのこととと、およろこびもうしあげます。さて・・・」というような長い文章を入力している途中で、文節の前後の単語のつながりを解析して自動変換する機能を「逐次変換機能」と言います。

▼句読点変換機能とは

「。」や「、」が入力された時点で、未変換の文節を自動変換する機能を「句読点変換機能」と言います。

▼キーファイルとは

キーファイルとは、AIかな漢字変換で使用する変換キーをカスタマイズ(自分用に変更)するためのファイルです。キーファイルはMS-DOS Ver.5.0をインストールした時点では存在しません。必要に応じてNECAIKEY(MS-DOSコマンド)で作成します。

▼CONFIG.SYSでの記述順位

AIかな漢字変換を登録する場合には、必ずKKCFUNC.SYS➡NECAIK1.DRV➡NECAIK2.DRVの順番で記述する必要があります。この3行のいずれかが抜けていると正常に動作しません。またHIMEM.SYSやEMM386.EXEなどのメモリマネージャより後に記述します。

▼コンベンショナルメモリの消費

AIかな漢字変換プログラムは、使用するメモリ環境によってコンベンショナルメモリの消費量が変化します。また同じEMSメモリを提供するEMM.SYSとEMM386.EXEでも、それぞれの消費メモリは変化します。結論を言うと、EMM386.EXEを使用するのがベストな選択です。

☆コンベンショナルメモリ消費量の違い

デバイスドライバ	EMS未使用	EMM.SYS	EMM386.EXE
NECAIK1.DRV	63,488	63,696	7,024
NECAIK2.DRV	61,088	368	368
合　計	124,576	64,064	7,392

※単位(バイト)

KKCFUNC.SYS *p.207*

プリント・システム

PRINT.SYS

●PC-PR系プリンタを制御する

NECのPC-PR系プリンタへの出力を制御できるようにします。

書　式

```
DEVICE=[d:][path]PRINT.SYS  [/U | /UL][/P1][/F]
```

パラメータ	設定内容
/U	起動時に外字ファイルをプリンタに送る (84文字以下)
/UL	起動時に外字ファイルをプリンタに送る (85文字以上)
/P1	全角文字を半角文字幅の1.5倍に設定する (印字比率の設定)
/F	プリンタをフルセントロニクス仕様で制御する (PC-H98のみ)

※/ULはMS-DOS Ver.5.0AのPRINT.SYSのみで使用可能

使用例

●プリンタドライバを登録する

```
DEVICE=PRINT.SYS
```

●起動時にプリンタへ外字ファイル (USKCG24.SYS) を登録する

```
DEVICE=PRINT.SYS /U    ←―― 84文字まで対応のプリンタ
DEVICE=PRINT.SYS /UL   ←―― 85文字以上対応のプリンタ
```

●全角文字と半角文字の印字幅比率を1.5対1に設定する

```
DEVICE=PRINT.SYS /P1
```

●PC-H98ノーマルモード使用時にフルセントロニクス仕様で制御する

```
DEVICE=PRINT.SYS /F
```

解　説

▼PRINT.SYSの実用性

PRINT.SYSはプリンタを制御するために用意されているプログラムです。しかし、一太郎やLotus1-2-3など多くのアプリケーションでは独自のプリンタ制御ルー

チンを内蔵しており、PRINT.SYS はほとんどが必要としません。主に MS-DOS コマンド (PRINT/COPY コマンドなど) で、テキストファイルをプリンタ出力する場合にのみに使用します。

また、PRINT.SYS で印字できるプリンタは、PC-PR 系またはそれをエミュレートするプリンタのみです。レーザープリンタのネイティブモードや NM 系プリンタなどへの出力はできません。

▼外字ファイルの使用

プリンタからユーザー作成の外字を印刷するには、24 ドットの外字ファイル (USKCG24.SYS) のドットパターンをあらかじめプリンタに送る必要があります。プリンタに外字ファイルを送るには、USKCGM (MS-DOS コマンド) から実行できます。

また PRINT.SYS に/U または/UL を指定しておくと、起動時に自動的にプリンタに外字ファイルが送られます。このとき外字ファイルは、起動ドライブのルートディレクトリに格納されている必要があります。

▼SWITCH コマンドでのプリンタ設定

PRINT.SYS を使用する場合、使用するプリンタの印字密度(16 ドット/24 ドット)は、あらかじめ SWITCH コマンドでパソコンに登録する必要があります。設定は「SWITCH ⏎」→「プリンタ ⏎」を選択します。

なお SWITCH コマンドでは、全角と半角の印字比率を設定する必要はありません。PRINT.SYS の印字比率設定が強制的に採用されます。

SWITCH コマンドでプリンタ仕様を設定する

```
SWITCHコマンド          Ver. 3.50
                         Copyright (C) NEC Corporation 1985,1991 -
プリンタ

        プリンタタイプ      24ドット系

        ANK/漢字           1/2

        設定終了

使用するプリンタのタイプを指定してください
矢印キー(↑・↓・←・→)で項目を選択し、リターンキーを押してください
(ESCキーを押すと前の画面に戻ります)
16ドット系  24ドット系
```

ラムディスク・シス

RAMDISK.SYS

●RAMディスクを使用できるようにする

メモリ上にRAMディスク（メモリ上に作成する一時的なディスクドライブ）を作成します。

書　式

```
DEVICE=[d:][path]RAMDISK.SYS  [ssss][bbbb][dddd][/M | /E]
```

パラメータ	設定内容
ssss	RAMディスク容量を128KB~14848KBの範囲で指定する（既定値は128）
bbbb	論理セクタ長を512または1024で指定する（既定値は1024）
dddd	ディレクトリ数を128KB~1024KBの範囲で指定する
/M	RAMディスクをコンベンショナルメモリ上に確保する
/E	RAMディスクをEMSメモリ上に確保する

使用例

●2MBのRAMディスクをEMBに確保する

```
DEVICE=RAMDISK.SYS 2048
```

●2MBのRAMディスクをEMSメモリに確保する

```
DEVICE=RAMDISK.SYS 2048 /E
```

●100KBのRAMディスクをコンベンショナルメモリに確保する

```
DEVICE=RAMDISK.SYS 100 /M
```

●論理セクタ長の指定

```
DEVICE=RAMDISK.SYS 2048 512        ←── 512バイト
DEVICE=RAMDISK.SYS 2048 1024       ←── 1024バイト
```

●ディレクトリ数の上限を設定する

```
DEVICE=RAMDISK.SYS 2048 1024 256   ←── 256個
DEVICE=RAMDISK.SYS 2048 1024 512   ←── 512個
```

解 説

▼RAM ディスクの用途

RAM ディスクとはメモリ上に作成されたディスクドライブです。コマンドやプログラムの一時作業ファイル（テンポラリファイル）の作成場所として使うことで、動作スピードを画期的に高速にできます。なお、/E で EMS メモリ内に RAM ディスクを確保する場合は、RAMDISK.SYS は EMS デバイスドライバより後に記述します。EMB に確保する場合は、HIMEM.SYS より後に記述します。

▼RAM ディスクの確保場所

/M または/E を付けないと、EMB（エクステンドメモリ）に RAM ディスクが確保されます。EMB 上に作成した RAM ディスクの方が、EMS に比べてアクセススピードは高速になります。なお、/M でコンベンショナルメモリに RAM ディスクを作成すると、他のアプリケーションが使用する領域が極端に少なくなります。通常は、/M は指定しないようにしてください。

▼ディレクトリ数の既定値

dddd オプションでディレクトリ数を指定しない場合は、次の表のようにディレクトリ数の最大値が決まります。「ディレクトリ数」とはドライブのルートディレクトリに作成できるファイル・ディレクトリの最大個数のことです。設定した数を超えて、ファイル・ディレクトリを作成することはできません。ただしサブディレクトリ内であれば、この制限は受けません。

RAMDISK 容量	ディレクトリ数	RAMDISK 容量	ディレクトリ数	RAMDISK 容量	ディレクトリ数
128KB〜1MB	128	5MB〜6MB	448	10MB〜11MB	768
1MB〜2MB	192	6MB〜7MB	512	11MB〜12MB	832
2MB〜3MB	256	7MB〜8MB	576	12MB〜13MB	896
3MB〜4MB	320	8MB〜9MB	640	13MB〜14MB	960
4MB〜5MB	384	9MB〜10MB	704	14MB〜14.5MB	1024

▼複数の RAM ディスクの同時作成

CONFIG.SYS には複数行の RAMDISK.SYS を指定することもできます。たとえば、2MB と 3MB の RAM ディスクを EMS メモリに作成するには、CONFIG.SYS に次の 2 行を追加します。この場合、記述した順番に若いドライブ番号が割り振られます。

```
DEVICE=A:¥DOS¥RAMDISK.SYS 2048
DEVICE=A:¥DOS¥RAMDISK.SYS 3072
```

アールエスディーアールブイ・シス

RSDRV.SYS

●RS-232C インターフェイスを制御する

RS-232C インターフェイスを制御できるようにします。

書　式

```
DEVICE=[d:][path]RSDRV.SYS
```

使用例

● RSDRV.SYS を登録する

```
DEVICE=RSDRV.SYS          ←─── コンベンショナルメモリに登録
DEVICEHIGH=RSDRV.SYS      ←─── UMB メモリに登録
```

解　説

▼RSDRV.SYS の実用性

RSDRV.SYS は、アプリケーションが RS-232C インターフェイスを利用してデータ通信を行うのを支援するプログラムです。しかし「まいと~く」などのメジャーな通信ソフトの多くは独自に RS-232C の制御ルーチンを持ち、RSDRV.SYS を使わないのが一般的です。現在では、MS-DOS コマンドの CTTY/COPY/COPYA などで RSDRV.SYS が必要なだけです。

▼RS-232C インターフェイスの初期化

MS-DOS コマンド（CTTY/COPY/COPYA など）で RS-232C インターフェイスを使用するには、事前に SPEED コマンドで RS-232C の初期化を実行する必要があります。これによって RSDRV.SYS 内部に持つ通信バッファへ、RS-232C からのデータ取り込みを開始できます。

また、RS-232C インターフェイスで通信するには、送信側と受信側の通信仕様（通信速度/ストップビット/キャラクタ長など）が同じでなければなりません。この通信仕様を設定するには SWITCH コマンドを使います。また SPEED コマンドで一時的に変更することも可能です。

セットバー・イグゼ

SETVER.EXE

●バージョンテーブルを変更する

アプリケーションがMS-DOSバージョンを参照しにきたら、偽ったバージョン番号を返答するよう設定します。特定のMS-DOSバージョンでないと起動しないアプリケーションを使用する場合に使用します。

書　式

DEVICE=[d:][path]SETVER.EXE

解　説

▼SETVER.EXEの用途

アプリケーションによっては起動時に特定のMS-DOSバージョンであるかどうかをチェックし、それに該当しない場合は起動しないものがまれにあります。SETVER.EXEはMS-DOS Ver.5.0が登場したばかりのときに、これらのチェックをクリアする目的で作成されました。しかしSETVER.EXEはあくまでバージョン番号を疑似的に変更するだけで、指定したDOSバージョンの機能をサポートするわけではありません。無理やりMS-DOSバージョンを偽って起動するため、アプリケーションによっては異常動作を引き起こす可能性もあるので、十分注意が必要です。

▼SETVER.EXEの使い方

プログラム「ABC.EXE」に対して、MS-DOS Ver.3.3を返すようにします。

①CONFIG.SYSに「DEVICE=A:¥DOS¥SETVER.EXE」を追加します。
②パソコンのリセットボタンを押し、MS-DOSを再起動します。
③コマンドラインから「SETVER ABC.EXE 3.30 ⏎」と入力します。
④パソコンのリセットボタンを押し、MS-DOSを再起動します。

これでABC.EXEは、起動中のMS-DOSバージョンを「MS-DOS Ver.3.3」であると錯覚して動作します。ただし、MS-DOS Ver.3.3の動作の安全性を保証するものでありません。

スマートドライブ・シス

SMARTDRV.SYS

●ディスクキャッシュを設定する

メモリ上にハードディスク用のディスクキャッシュを作成します。ハードディスクを高速にアクセスできるようになります。

書 式

DEVICE=[d:][path]SMARTDRV.SYS [サイズ1[サイズ2]][/E]

パラメータ	設定内容
サイズ1	標準キャッシュ容量を128~8192から指定（既定値は256）
サイズ2	メモリ競合時の最小キャッシュ容量を128~8192から指定（既定値は0KB）
/E	EMSメモリ上にディスクキャッシュを確保する

使用例

- 2MBのディスクキャッシュをEMSメモリに確保する
  DEVICE=SMARTDRV.SYS 2048 /E
- 2MBのディスクキャッシュをEMBメモリに確保する
  DEVICE=SMARTDRV.SYS 2048
- メモリ競合時に256KBまで使用容量を縮小する
  DEVICE=SMARTDRV.SYS 2048 256

解 説

▼SMARTDRV.SYSの特徴

SMARTDRV.SYSは、読み込み命令のあったデータもそうですがそれより先のデータもあらかじめ読み込みます（先読み機能）。したがってシーケンシャルアクセスをする場合には画期的な効力を発揮します。また、他のアプリケーションと使用メモリが競合した場合は、状況に応じてメモリを明け渡し、特にWindows使用時にはメモリ資源の有効活用ができます。なお、ディスクキャッシュ容量は、16KBの倍数で指定すると効率的にメモリを使うことができます（例:1536, 2048, 2560, 3072……）。

参照 HIMEM.SYS *p.206* EMM386.EXE *p.201*

第３章

その他のデバイスドライバ一覧

市販されているポピュラーなメモリマネージャや日本語入力プログラム（FEP）などの、デバイスドライバを各アプリケーションごとに解説します

エィトック８エイ（ビー/イーエックス）・シス	一太郎 Ver.5

ATOK8A.SYS
ATOK8B.SYS
ATOK8EX.SYS

●ATOK8 で日本語入力をできるようにする

ATOK8 の動作環境を設定するデバイスドライバです。

書　式

```
DEVICE=[d:][path]ATOK8A.SYS  [/UCF=d:path][/M=n]
        [/KC=n][/B=n][/S=n][/SD=nnnn][/O=n][/C=n]
        [/E=n][/T=nnnn][/DM=n][/Dn=d:path][/I=n]
        [/G=d:path][/SP=nnn][/Z=1]
DEVICE=[d:][path]ATOK8B.SYS
DEVICE=[d:][path]ATOK8EX.SYS
```

パラメータ	設 定 内 容
/UCF	環境設定ファイル（ATOK8.UCF）のあるパスを設定する
/M	漢字入力モードを設定する（0:R 漢　1:カナ漢）
/KC	入力文字種を設定する（0:あ　1:ア　2:ｱ　3:A　4:A）
/B	変換モードを設定する（0:連文節　1:単文節　2:自動）
/S	辞書学習モードを設定する（0:しない　1:する）
/SD	自動登録を４桁（後変換・未登録・複合語・文節区切）で設定する （0:しない　1:する）　例：/SD=1101
/O	送り仮名を設定する（0:本則　1:省く　2:送る）
/C	コード体系を設定する（0:JIS　1:シフト JIS　2:区点）
/E	入力位置を設定する（0 システムライン　1:エコー）

パラメータ	設定内容
/T	句読点モードを4桁で指定する (0:, 1:、) (0:. 1:。) (0:[1:「) (0:/ 1:・)
/DM	基本辞書番号を0~9で指定する
/Dn	辞書ファイル番号ごとのパスを指定する
/I	外字ファイルの使用を設定する (0:しない 1:する)
/G	外字ファイルのある場所を設定する
/SP	文字種 (あアｱAA) の順でスペースキーの機能を制御する (0:空白 1:機能)
/Z=1	DPMI環境下でATOK8を使用する場合に、動作を安定させる

使用例

● 一太郎付属のメモリマネージャでATOK8を使う

```
DEVICE=A:¥ATOK8¥EMS386.SYS /X /I
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
```

● DPMI規格管理下でATOK8を安定動作させる

```
DEVICE=A:¥MDEV¥ISOPRO¥VMM386.EXE /I /U  ←—— MemoryServer II
DEVICE=A:¥DOS¥KKCFUNC.SYS
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF /Z=1
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICE=A:¥ATOK8¥ATOK8EX.SYS
```

● Bドライブにある ATOK8 環境ファイルを使用する

```
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=B:¥ATOK8.UCF
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICEHIGH=A:¥ATOK8¥ATOK8EX.SYS
```

● 漢字入力モードの初期値をカナ漢字変換にする

```
DEVICE=A:¥ATOK8¥ATOK8A.SYS /UCF=A:¥ATOK8¥ATOK8.UCF /M=1
DEVICE=A:¥ATOK8¥ATOK8B.SYS
DEVICEHIGH=A:¥ATOK8¥ATOK8EX.SYS
```

解　説

▼ATOK8A.SYSのパラメータについて

通常ではATOK8はMS-DOS起動時に、ATOK8環境ファイル「ATOK8.UCF」の内容を読み込み、その内容に沿って入力環境を構築します。しかし、CONFIG.SYSのATOK8A.SYSにオプションを付けて起動すると、ATOK8環境ファイルの設定内容より、CONFIG.SYSでの設定の方が優先されます。

▼ATOK8動作環境の設定方法

CONFIG.SYSにオプションを設定する以外にATOK8の動作環境を設定するには、環境設定ユーティリティ（ATUT.EXE）またはジャストウィンドウ（JW2）のJW環境設定で行う方法があります。設定内容はいずれもATOK8.UCFに保存されます。

ATUT.EXEは、ATOK8が付属する製品に含まれるユーティリティです。

イーエムエス 386・シス　　一太郎 Ver.5

EMS386.SYS

●VCPI 規格でプロテクトメモリを管理する

MS-DOS 版一太郎 Ver.5/花子 Ver.3/三四郎/五郎に付属する VCPI 規格対応のメモリマネージャです。i80386/i80486CPU を搭載したパソコンで最大 16MB まで管理します。

書　式

```
DEVICE=[d:][path]EMS386.SYS  [/3 | /4]
        [/W=XX, XX, XX, XX | /Q=XX, XX, XX, XX, XX, XX, XX, XX]
        [/X][/I]
```

パラメータ	設定内容
/3	EMS 規格のバージョンを 3.XX に設定する
/4	EMS 規格のバージョンを 4.XX に設定する (既定値)
/W	指定したアドレスに強制的に EMS 物理ページを設定する
/Q	指定したアドレスの空領域に EMS 物理ページを設定する
/X	64KB のハイメモリを利用できるようにする
/I	起動時にメモリ設定情報を表示する

使用例

●ハイメモリ (HMA) を利用できるようにする

```
DEVICE=A:¥ATOK8¥EMS386.SYS /X
```

●EMS 物理ページを任意のアドレスに強制的に設定する

```
DEVICE=A:¥ATOK8¥EMS386.SYS /W=C0, C4, C8, CC
```

●EMS 物理ページを指定アドレスの空領域に確保する

```
DEVICE=EMS386.SYS /Q=B0, B4, B8, BC, C0, C4, C8, CC
```

●起動時にメモリ設定情報を表示する

```
DEVICE=A:¥ATOK8¥EMS386.SYS /I
```

●EMS バージョンを指定する

```
DEVICE=A:¥ATOK8¥EMS386.SYS /3 ◀──── LIM EMS 3.X
DEVICE=A:¥ATOK8¥EMS386.SYS /4 ◀──── LIM EMS 4.X
```

解 説

▼サウンド BIOS の切り離し

98 シリーズを使う場合、パソコン本体のサウンド BIOS が「使用する」に設定してあると、EMS メモリが確保されません。「サウンド BIOS」とは、パソコンに内蔵されている FM 音源・SSG 音源などのサウンド機能の制御プログラムです。このサウンド機能は一部のゲームソフトでしか使用しないので、通常は切り離しておいても問題はありません。切り離しは次の手順で行います。

①HELP キーを押しながらリセットボタンを押します。
②セットアップメニューが表示されるので、ROLL UP キーを押し、「サウンド機能」の項目があるページを表示します。
③「切り離す」にカーソルを移動し、ESC キーを押します。

▼PC-H98/98MATE シリーズでの使用

PC-H98 シリーズで使用する場合は、ハードディスクの DMA チャネルを「3」に設定してください。また、PC-H98 model105 で使用する場合は、内蔵ハードディスクは「互換モード」を選択する必要があります。

PC-H98 や 98MATE をハイレゾリューションモードで起動する場合は、あらかじめ SWITCH コマンドでコンベンショナルメモリを 640KB に設定する必要があります。

▼MS-DOS Ver.5.0A のメモリマネージャとの共存

EMS386.SYS は、MS-DOS Ver.5.0A の HIMEM.SYS と同時に CONFIG.SYS に指定することはできません。また SMARTDRV.SYS や RAMDISK.SYS などのメモリを活用するデバイスドライバも同様に併用できません。必ず EMSDISK.SYS や EMSCACHE.SYS を使用してください。

▼ジャストシステム製メモリ拡張ボードの使用

ジャストシステム製メモリ拡張ボードを使用している場合は、ボードを「リニアモード」に設定してください。

参照 **EMSDISK.SYS** *p.227* **EMSCACHE.SYS** *p.228*

EMSDISK.SYS

●メモリ上にRAMディスクを設定する

EMS386.SYSで確保したEMSメモリ上に指定した容量のRAMディスクを設定します。

書　式

```
DEVICE=[d:][path]EMSDISK.SYS  /K=nnnnnK
```

パラメータ	設定内容
/K	RAMディスクとして確保するメモリ容量をKB単位で指定する

使用例

●RAMディスクに2MB（2048KB）を割り当てる

```
DEVICE=EMSDISK.SYS /K=2048K
```

解　説

▼RAMディスクとは

RAMディスクとはメモリ上に作るドライブのことで、ハードディスクなどに比べて高速なデータの読み書きが可能です。98シリーズでは、RAMディスクは接続されているドライブの最終ドライブ名が割り振られます。

EMSDISK.SYSは必ずEMS386.SYSより後に記述してください。また、他のメーカーのメモリマネージャとは共存できません。

▼RAMディスクの記憶内容

パソコンのリセットボタンを押したり電源をOFFにすると、RAMディスクの記憶内容は消去されます。したがってRAMディスクには、突然の停電やパソコンのハングアップなどの際に失っては困るようなデータを格納することは、避けた方がよいでしょう。

📖 **EMS386.SYS** *p.225*　**EMSCACHE.SYS** *p.228*

イーエムエスキャッシュ・シス　　　　　　　　　一太郎 Ver.5

EMSCACHE.SYS

●メモリ上にディスクキャッシュを設定する

EMS386.SYS で確保した EMS メモリ上に、指定した容量のディスクキャッシュを設定します。なお、書込みデータはキャッシュ対象になりません。

書 式

```
DEVICE=[d:][path]EMSCACHE.SYS  /K=nnnnK  [/D=nnnn]
       [/X=nnnnnn][/H=nnn][/I]
```

パラメータ	設 定 内 容
/K	ディスクキャッシュに確保する容量を128K~8192KからKB単位で指定する
/D	キャッシュ対象とするフロッピーディスク装置番号を1~4の範囲で指定する
/X	キャッシュ対象とする SCSI 型 HDD の ID 番号を 1~6 の範囲で指定する
/H	キャッシュ対象とする SCSI 以外の HDD 装置番号を指定する
/I	起動時にディスクキャッシュ設定情報を表示する

使用例

●1 台目の SCSI ハードディスクに 2MB (2048KB) のキャッシュを設定する

```
DEVICE=EMSCACHE.SYS /K=2048K /X=1
```

●1台目のIDEハードディスク装置に2MB(2048KB)のキャッシュを設定する

```
DEVICE=EMSCACHE.SYS /K=2048K /H=1
```

●2 基のフロッピーディスク装置に 512KB のキャッシュを設定する

```
DEVICE=EMSCACHE.SYS /K=512K /D=12
```

解 説

▼使用時の注意点

モーターが一定時間後に停止するタイプの FDD 装置を採用する機種 (98NOTE シリーズ、PC-386AR など)、または 1.44MB のフロッピーディスクドライブは、EMSCACHE.SYS のキャッシュ対象にできません。また、NEC ハードディスクインターフェイスに完全互換でない場合は、EMSCACHE.SYS の対象にできない場合があります。PC-H98model60/70/100 の内蔵ハードディスクを/H オプションで指定するには、必ず「/H=3」を指定してください。

参照　EMS386.SYS *p.225*　EMSDISK.SYS *p.227*

VMM386.EXE

●VCPI規格でプロテクトメモリを管理する

最大64MBまでのプロテクトメモリを、VCPI規格に準拠して管理します。一太郎Ver.5とWindowsを共存させることができます。

書 式

```
DEVICE=[d:][path]VMM386.EXE  [/E | /NE][/F=XX]
        [/W=XX-XX][/3 | /4 | /J[=X]][/EH=nnn][/X | /NX]
        [/H=nn | /NH][/XH=nnn][/U[=XX] | /NU][/V | /NV]
        [/CA][/ROM=XX-XX, …][/HROM=XX-XX, …]
        [/@=EMM386.SYS][/M[=XX:YY-ZZ][/NECID][/I]
        [/B=nnn][/M[=xx:y]][/T=EXTDSWAP.SYS]
```

パラメータ	設定内容
/E	EMS (LIM EMS4.0) を使用する (既定値)
/NE	EMSを使用しない
/F	EMS物理ページを割り当てる、先頭アドレス (C0/C4/C8/CC/CD) を設定する
/W	強制的に割り当てる、EMS物理ページアドレスを設定する
/3	LIM EMS3.2互換モードに設定する
/4	LIM EMS4.0互換モードに設定する (既定値)
/J	拡張LIM EMS4.0モードに設定する(ウィンドウ数を1~4の範囲で指定可能)
/EH	EMSのハンドル数を16~128の範囲で指定する (既定値は64)
/X	XMSを使用する (既定値)
/NX	XMSを使用しない
/H	HMAを使用 (HMAサイズを0~64から指定可能) する (既定値)
/NH	HMAを使用しない
/XH	HMAを使用しない
/U	UMBを使用 (アドレスを16進数4KB単位で指定) できるようにする

パラメータ	設定内容
/NU	UMB を使用しない (既定値)
/V	VCPI を使用する (既定値)
/NV	VCPI を使用しない
/CA	Cx486DLC のキャッシュを ON にする
/ROM	EMS ページフレームや UMB に割り当てを禁止する領域を、16 進数 4KB 単位で指定する
/HROM	ハイレゾモード使用時に UMB に割り当てを禁止する領域を、16 進数 4KB 単位で指定する
/@	ハイレゾモード使用時に指定する (コンベンショナルメモリ 758KB に対応させる)
/M	SCSI または SASI ハードディスクの BIOS ROM を空領域に移動する (/M だけのときは自動サーチして BIOS を移動する) XX　移動元の ROM アドレスセグメント上位 2 バイトを指定する YY　移動する ROM 容量を指定する ZZ　移動先の ROM アドレスセグメント上位 2 バイトを指定する
/NECID	UMB で BIOS をふさいだとき、Windows 起動時の NEC-ID チェックをクリアし、起動できるように設定する
/I	起動時に VMM386.EXE のメモリ設定情報を表示する
/B	仮想 I・O バンクメモリ使用時のサイズを 10 進数で指定する
/T	DOSSHELL の拡張タスクスワップ対応ドライバを組み込む

使用例

● XMS (HMA/UMB/EMB) を使用できるようにする

```
DEVICE=VMM386.EXE /X /H /U
```

● EMS ハンドル数を 16 に変更しメモリを節約する

```
DEVICE=VMM386.EXE /EH=16
```

● サウンド BIOS 上に強制的に EMS ページフレームを設定する

```
DEVICE=VMM386.EXE /W=CC /F=C0 ← C0000h~CFFFFh (64KB) に割り当てる
```

● UMB を利用する

```
DEVICE=VMM386.EXE /U ←—— C0000h~DFFFFh の空き領域を UMB にする
DEVICE=VMM386.EXE /U=D0-D7 ← D0000h~D7000h を強制的に UMB にする
DEVICE=VMM386.EXE /U=D0-D7, E8-F4 ←—— 複数の領域を強制的に UMB にする
```

●ハードディスクの BIOS を移動し UMB を広くする

DEVICE=VMM386.EXE /U /M ←― SCSI/SASI の ROM を A5000h に移動する

DEVICE=VMM386.EXE /U /M=D8:16-F0 ←― IDE の ROM を F0000h に移動する

●NEC-ID を UMB でふさいだときに Windows を起動できるようにする

DEVICE=VMM386.EXE /U=D0-D7, E8-F4 /NECID

●起動時にメモリ設定情報を画面に表示する

DEVICE=VMM386.EXE /I

●MS-DOS Ver.5.0A の拡張タスクスワップ対応ドライバを組み込む

DEVICE=VMM386.EXE /T=A:¥DOS¥EXTDSWAP.SYS

解 説

▼VMM386.EXE の特徴

MemoryServer II になって、メモリマネージャのファイル名は VMM386.EXE に変更されました（前バージョンの MemoryServer では VMM386.SYS)。主な特徴は次の 3 点です。

- ●64MB までのプロテクロメモリを管理可能
- ●MS-DOS Ver.5.0A の拡張タスクスワップ対応ドライバに対応
- ●ハードディスク（SASI/SCSI/IDE）の BIOS を移動可能

なお、前バージョンでは DPMI サーバが内蔵されていましたが、VMM386.EXE から切り離されています。そのかわり DPMI32.EXE という別プログラムがあり、コマンドラインから実行することで DPMI サーバを起動できます。

▼ハードディスク BIOS の移動 注意

/M オプションでアドレス記述を省略（/M のみ）すると、UMA 内にあるハードディスク BIOS (SASI 型及び SCSI 型の双方が対象）を自動的に A5000h~A7FFFh (12KB) に移動します。これにより連続した UMB を広く取ることができます。IDE 型ハードディスクの BIOS を移動するには、「/M=D8:16-F0」などのように指定します。この指定は「D8000h から 16KB の BIOS を F0000h に移動する」という意味です。

なお、/M は移動先のアドレスが正しいかどうかをチェックしません。使用されているアドレス上に重複して指定しないように注意が必要です。またアイ・オー・データ機器のシリコンディスク（SDB シリーズ）や 98NT II 使用時には、/M は使用できません。

▼UMA上のBIOS ROMのアドレス 参照 p.85

UMAのどの位置にBIOS（ハードディスクなどハードウェアを制御するROM）が割り当てられているかは、それぞれ使用するパソコン機種や周辺機器機種によって異なります。一般的にはIDE型ハードディスクはD8000h~DBFFFh、SCSI型ハードディスクはDC000h~DCFFFh、SASI型ハードディスクはD7000h~D7FFFhにBIOSが割り当てられています。E8000h~F4FFFはBASIC BIOSが割り当てられています。なお、ノート型パソコンでは次の領域は常にハードディスクやRAMドライブのBIOSで埋められているため、UMBに設定できません。

機　種	アドレス範囲
98NOTEシリーズ	D8000h~DBFFFh
PC-386NOTE AE/AR/W/WR	D8000h~DBFFFh
PC-386NOTE A/BookL	D4000h~DFFFFh
PC-486NOTE AS	D8000h~DBFFFh

DPMI32.EXE

●DPMIサーバを起動する

VCPIを利用してDPMIサーバを起動します。コマンドラインとデバイスドライバの双方の形式で使用することができます。

書　式

Ⅰ. **DEVICE=[d:][path]DPMI32.EXE　[/S=XX][/L=XXXX]**
[[/X | /NX] | [/V | /NV]]
Ⅱ. **DPMI　[/S=XX][/L=XXXX][/U | /NU][/ON | /OFF][/R]**

※Ⅰはデバイスドライバ形式/Ⅱはコマンド形式のときの書式

パラメータ	設定内容
/S	ストリームバッファの容量を1~64の範囲で1KB単位で指定する(既定値は4)
/L	DPMIが提供するLDT数を512~8192の範囲で指定する（既定値は8192）
/X	DPMIのメモリ資源をXMSメモリから調達する
/NX	DPMIのメモリ資源をXMSメモリから調達しない（既定値）
/V	DPMIのメモリ資源をVCPIから調達する（既定値）
/NV	DPMIのメモリ資源をVCPIから調達しない
/U	UMBをストリームバッファに利用する（DOS=UMB指定時のみ）
/NU	UMBをストリームバッファに利用しない（既定値）
/ON	/OFFで停止したDPMIサーバを有効にする
/OFF	DPMIサーバを一時的に停止する
/R	DPMI.EXEの常駐を解除する

使用例

●CONFIG.SYSからDPMIサーバを起動する

```
DEVICE=DPMI32.EXE
```

●コマンドラインからDPMIサーバを起動する

```
DPMI32 ↵
```

解　説

DPMI32.EXE は、コマンドラインと CONFIG.SYS のどちらから実行しても同じ機能です。ただし、コマンドラインから実行すると、不要なときにメモリ上から解除できるという特徴があります。MS-DOS の子プロセスからの実行はできません。

また、PC-386NOTE A で DPMI サーバを起動する際には、レジューム機能は使用できなくなります。

IOS10.EXE

●メモリ上にRAMディスクを設定する

VMM386.EXEで確保したメモリ上にRAMディスクに設定します。

書 式

I. DEVICE=[d:][path]IOS10.EXE Size [/E | /X | /B][/S | /NS]
[/C=nn][/D=nnnn][/L=nnnn][/Q | /NQ][/V | /NV]
[/A=xxyyz][/W | /NW][/R=d][/M][/16 | /32]
[/A=yyxxz][/W | /NW]

II. IOS10 [/S | /NS][/C=nn][/D=nnnn][/L=nnnn][/A=xxyyz]
[/W | /NW][/Y=nnnnn][/F][/K[=XXX]]

III. IOS10 [/?]

※ I はデバイスドライバ、II/III はコマンドラインでの書式

パラメータ	設定内容
Size	RAMディスク容量を128KB~32768KBの範囲で指定する
/E	RAMディスクをEMS上に作成する
/X	RAMディスクをEMB上に作成する
/B	RAMディスクをBMS(バンクメモリ)上に作成する
/S	チェックサムかHiSUMによるデータ検査を行う(既定値)
/NS	データ検査を行わない
/C	クラスタサイズを1/2/4/5/16のいずれかで指定する(既定値は2セクタ)
/D	ディレクトリ数を96~2048から32個単位で指定する(既定値は192)
/L	セクタサイズを256/512/1024のいずれかを指定する(既定値は512)
/Q	起動時にRAMディスクの初期化を確認する
/NQ	起動時にRAMディスクの初期化を確認しない(既定値)
/V	起動時にRAMディスクを初期化する(既定値)
/NV	起動時にRAMディスクを初期化しない (ただしプロテクトメモリ使用時には常に初期化される)
/A	RAMディスクのアクセスランプを表示する xx ライン座標を00~24の範囲で指定する yy カラム座標を00~79の範囲で指定する(FNと指定するとシステムライン) z 表示色を0~7から指定する(黒/赤/緑/黄/青/紫/シアン/白)

パラメータ	設定内容
/W	RAM ディスクを書込禁止にする
/NW	RAM ディスクを書込禁止にしない (既定値)
/R	RAM ディスクにドライブ名を割り当てる (カレント空きドライブ~Z の範囲)
/M	DOS シェル使用時に RAM ディスクを認識させる
/16	常に RAM ディスクを 16 ビット幅でデータ転送する
/32	32 ビット CPU では常に 32 ビット幅でデータ転送する (既定値)
/Y	変更する RAM ディスク容量を 128~32768KB の範囲で指定する
/F	RAM ディスクの初期化を行う
/K	RAM ディスクのドライブ名を指定した環境変数に格納する
/?	コマンドヘルプを表示する

使用例

- RAM ディスクに 2MB (2048KB) を割り当てる

```
DEVICE=IOS10.EXE 2048 /X    ←── EMB 上に作成
DEVICE=IOS10.EXE 2048 /E    ←── EMS 上に作成
DEVICE=IOS10.EXE 2048       ←── EMS→BMS→EMB の順に空いている先に作成
```

- DOSSHELL で RAM ドライブを認識できるようにする

```
DEVICE=IOS10.EXE 2048 /M
```

- クラスタサイズを 16 にする

```
DEVICE=IOS10.EXE 2048 /C=16
```

- アクセスランプをシステムライン右端に黄色で表示する

```
DEVICE=IOS10.EXE 2048 /A=FN793
```

- コマンドラインから実行する

```
IOS10 /Y=1024 ↵    ←── 容量を 1024KB に変更
IOS10 /M ↵         ←── ドライブ名を環境変数 TMP に格納
```

解　説

▼複数の RAM ディスクの設定

IOS10.EXE で複数の RAM ディスクを設定するには、CONFIG.SYS に連続して登録します。最大 8 個まで登録できます。ドライブ名は登録した順に割り当てられます。

▼プロテクトメモリの使用

プロテクトメモリを使用している場合は、/NV オプションを付けても RAM ディスクはリセット時に初期化されます。

VMM386.EXE *p.229*　DC10.EXE *p.237*

DC10.EXE

●キャッシュディスクを設定する

VMM386.SYS で確保したメモリ上にディスクキャッシュに設定します。

```
I. DEVICE=[d:][path]DC10.EXE  Size  [d:][/E | /X | /B]
          [/S | /NS][/V | /NV][/W=nnnn][/A | /NA]
          [/L | /NL][/F | /NF][/C | /NC]
II. DC10  [/A | /NA][/L | /NL][/F | /NF][/C | /NC][/H | /NH]
          [/D | /ND][/@=Size (nnnnn) ][/@=RESTART (d) ]
          [/@=REFRESH (d) ][/@=LOCFILE (ファイル名) ]
          [/@=READFAT (d) ][/@=READDIR (d) ]
III. DC10 [/? | /@]
```

※ I はデバイスドライバ、II/III はコマンドラインでの書式

パラメータ	設定内容
Size	キャッシュサイズを 256KB~16384KB の範囲で指定する
/E	EMS 上にキャッシュバッファを確保する
/X	EMB 上にキャッシュバッファを確保する
/B	BMS (バンクメモリ) 上メモリにキャッシュバッファを確保する
/S	プログラムの一部 (データ部) を EMS に置く
/NS	プログラムの一部 (データ部) をコンベンショナルメモリに置く (既定値)
/V	ディスク BIOS へのキャッシュを有効にする (/E、/X 指定時の既定値)
/NV	ディスク BIOS へのキャッシュを無効にする (/B 指定時の既定値)
/W	Windows 使用時のキャッシュサイズを 0~16384 から 128KB 単位で指定する
/A	フロッピー入れ替え時に警告音を出す (既定値)
/NA	フロッピー入れ替え時に警告音を出さない
/L	キャッシュした FAT 情報をロックし、アクセスを高速化する (既定値)
/NL	キャッシュした FAT 情報をロックしない
/F	バッファ満杯時に掃き出しを行う (既定値)

パラメータ	設定内容
/NF	バッファ満杯時に掃き出しを行わない
/C	対象ドライブのキャッシングを許可する（既定値）
/NC	対象ドライブのキャッシングを許可しない
/H	キャッシュしたデータを掃き出し禁止にする（辞書などに使う）
/NH	/H の指定を解除する
/D	ディレクトリ情報をメモリ上に保持する
/ND	/D で保持したディレクトリ情報を解除する
/@=Size	キャッシュサイズを指定した容量に変更する
RESTART	指定ドライブのキャッシュデータを破棄する
REFRESH	指定ドライブの/D、/L、/H で保持した以外のデータを破棄する
LOCFILE	指定したファイルをキャッシュし、掃き出し禁止にする
READFAT	指定ドライブの FAT 情報をキャッシュする
READDIR	指定ドライブのディレクトリ情報をキャッシュする
/?	コマンドヘルプを表示する
/@	/@が付くオプションのコマンドヘルプを表示する

使用例

●キャッシュバッファに 2MB（2048KB）を割り当てる

DEVICE=DC10.EXE 2048 ←―― EMS→BMS→EMB の順に空いている先に作成

DEVICE=DC10.EXE 2048 /X ←―― EMB 上に作成

●MS-DOS 実行時には 2048KB、Windows 実行時には 3072KB を割り当てる

DEVICE=DC10.EXE 2048 /W=3072

●A ドライブと B ドライブをキャッシュ対象にする

DEVICE=DC10.EXE A: B: 2048

●コマンドラインから実行する

DC10 /H ↵ ←―― 掃き出し禁止にする

DC10 /D ↵ ←―― 最新のディレクトリ情報をキャッシュする

DC10 /@=Size (1024) ↵ ←―― キャッシュ容量を 1024KB に変更する

DC10 ↵ ←―― 現在のキャッシュ情報を表示する

解　説

▼キャッシュを設定するメモリ種別

DC10.EXEはEMS/XMS(EMB)/BMSの3種類のメモリ上に設定できます。ディスクキャッシュはEMB上（/Xオプション）に作成した方がアクセススピードは高速で、かつ細かく設定できます。ちなみに各メモリ種別のメモリ設定の最小単位は、次のようになります。

メモリ	最小単位
EMB上に作成する	8KB単位
EMS上に作成する	16KB単位
BMS上に作成する	128KB単位

▼キャッシュできないドライブ

RAMディスク/MO（光磁気ディスク）/CD-ROM/2DDフロッピーディスク/圧縮されたドライブには、キャッシュを設定できません。ただし、DiskXⅡ(A・I・Soft)で作成した圧縮ドライブはキャッシュできます。

▼UMBへのロードについて

ノート型パソコンでDC10.EXEをUMBにロードする場合、フロッピーディスクドライブがキャッシュ対象になっていると、レジューム機能がうまく動作しない場合があります。

▼フロッピーディスクに対するキャッシュ

フロッピーディスク装置をキャッシュ対象にすると、ドライブのモーターが常時回転し、電力消費量が増えます。したがって、ノート型パソコンではバッテリーの消耗が早くなる場合があります。またヘドアンロード機能（非アクセス時にヘッド位置を退避する機能）がない装置では、フロッピーディスクが摩耗する場合があるので注意が必要です。

▼光磁気ディスクへのキャッシュ設定

NEC純正ユニットで読み書きできるメディア（IBMまたはセミIBMフォーマット）以外では、正常に動作しない場合があります。

VMM386.EXE *p.229*　**IOS10.EXE** *p.235*

LUMB.EXE

●UMBに常駐プログラムをロードする

デバイスドライバやTSR（常駐プログラム）をUMBにロードします。コマンドラインからの実行とCONFIG.SYSへの登録との2つの方法があります。

書　式

Ⅰ. **DEVICE=[d:][path]LUMB.EXE　[/INFO][/M][/S=hhhhh]**
[/B=d | /BN=d | /BH=d]　デバイスドライバ名
Ⅱ. **LUMB [[/MAP][/I][/F][/NOESC] | [/?]]**
Ⅲ. **LUMB [/INFO][/M][/S=hhhhh][/ENVLAST]　TSR名**

※Ⅰは CONFIG.SYS での書式、ⅡとⅢはコマンドラインでの書式

パラメータ	設定内容
/INFO	起動時にUMBの詳細情報を表示する
/M	UMBメモリ不足時にはコンベンショナルメモリにロードする
/S	ロード時に必要とするメモリ容量を16進数5桁で指定する
/B	指定するUMBブロック番号（d）にロードする
/BN	ノーマルモード使用時に指定するUMBブロック番号（d）にロードする
/BH	ハイレゾモード使用時に指定するUMBブロック番号（d）にロードする
/MAP	UMBマップを表示する
/I	UMBのブロック情報を表示する
/F	UMBのフリーエリア情報を表示する
/NOESC	UMBマップをテキストモード（モノクロ）で表示する
/?	コマンドヘルプを表示する
/ENVLAST	環境エリアをUMB最上位に作成する

使用例　p.129

●PRINT.SYSを1番目のUMBブロックにロードする（CONFIG.SYS）

```
DEVICE=A:¥IMDEV¥IOSPRO¥LUMB.EXE /M /B=1 A:¥IDOS¥PRINT.SYS
```

●DOSKEY コマンドを UMB にロードする

LUMB /M DOSKEY /INSERT⏎

●FASTOPEN をロードする際に 16368 バイト (3FF0) をあらかじめ確保する

LUMB /M /S=03FF0 FASTOPEN A: B:⏎

●UMA のどの領域に何があるかを調べる (UMB マップ)

LUMB /MAP⏎

●UMB の使用状況を調べる (UMB リスト)

LUMB /I⏎

●LUMB の使い方を調べる

LUMB /?⏎

解 説

▼LUMB.EXE の特徴

LUMB.EXE は MS-DOS Ver.5.0A に付属する DEVICEHIGH (CONFIG.SYS コマンド) と LOADHIGH (MS-DOS コマンド) の機能を合わせ持ったコマンドです。LUMB.EXE の優れている点は、CONFIG.SYS で指定する際にロードする UMB ブロックを指定できる点にあります。

ただし、残念ながらコマンドラインから TSR をロードする際には、UMB ブロックを指定できません。なお、LUMB.EXE はあらかじめ VMM386.EXE で UMB が確保 (/U) されていないと、使用できません。

▼LUMB 使用時の注意点

LUMB を使用する際には、次の点に注意してください。

DEVICEHIGH と混在させない

CONFIG.SYS 内で LUMB と DEVICEHIGH を混在して使用すると、正しくロードできない場合があります。

ロードの際には/M を付ける

デバイスドライバ (または TSR) をロードする際には、必ず/M オプションを付けてください。/M を付けると UMB へのロードが失敗したときに、コンベンショナルメモリに自動的にロードしてくれます。

ロードが不可能なデバイスドライバ

構造的に 11 個以上のデバイスエントリを持つデバイスドライバは、LUMB コマンドで UMB にロードできません。

参照 DEVICEHIGH *p.189*

MELMM.386

●VCPI 規格でプロテクトメモリを管理する

最大64MBまでのプロテクトメモリを、VCPI規格に準拠して管理します。一太郎Ver.5とWindowsを共存させることができます。

書　式

DEVICE=[d:][path]MELMM.386　[/CX][/HM][/M nn, nn…]
[/NC][/Pn][/SD nnnn][/SW1][/T][/XMS nnnn]
[/BEnnnn][/H][/NECWIN]

パラメータ	設定内容
/CX	Cx486DLC/SLC用のキャッシュコントローラを動作させる
/HM	XMSを利用する（同じディレクトリにMELMM.VXDが必要）
/M	UMB領域にするアドレスの先頭2桁を指定する
/NC	EMSの一部のファンクション命令の動作を高速にする
/Pn	EMS物理ページ数を2、3、8の中から指定する（既定値は4）
/SD	EMS物理ページを割り当てる先頭アドレス（セグメント）を指定する
/SW1	D000-DFFFのSCSI/SASIハードディスクのROM BIOSをA500-AFFFに移動する
/T	MS-DOS Ver.5.0Aの拡張タスクスワップを利用する
/XMS	XMSとして管理するメモリ上限をKB単位で指定する
/BE	プロテクトメモリをバンクメモリにするときの、バンク数(128KB単位)を指定する
/H	EMSのハンドル数を2~255の範囲で指定する（既定値は64）
/NECWIN	UMBでBIOSをふさいだとき、Windows起動時のNEC-IDチェックをクリアし、起動できるように設定する

使用例

●XMSとEMSを使用する

DEVICE=MELMM.386 /HM

●D0000h~D3FFFhとD4000h~D7FFFhをUMBにする

DEVICE=MELMM.386 /HM /M D0, D4

●SCSI/SASI 型のハードディスク BIOS を A5000h 以降に移動する

```
DEVICE=MELMM.386 /SW1 ←── BIOSの移動
DEVICE=MELMM.386 /HM /SW1 /M D0,D4,D8,DC ←─ 移動した場所をUMBにする
```

●MS-DOS Ver.5.0A の DOS シェルに対応させる

```
DEVICE=MELMM.386 /HM /XMS 2048 /T A:¥DOS¥EXTDWAP.SYS
```

●EMS の一部のファンクションを高速に処理する

```
DEVICE=MELMM.386 /HM /NC
```

●Cx486DLC/SLC を使用できるようにする

```
DEVICE=MELMM.386 /CX
```

解　説

▼ハードディスク BIOS の移動

/SW1 を指定すると、D0000h~DFFFFh にある SASI 型と SCSI 型ハードディスクの ROM BIOS は、自動的に A5000h~AFFFFh に移動します。ただし IDE 型ハードディスク（通常は D8000h~DBFFFh にある）は移動の対象にはなりません。また 98 NOTE では D8000h~DBFFFh はハードディスクや RAM ドライブの BIOS が割り当てられています。したがって/SW1 を指定すると正常に動作しなくなるので、注意が必要です。

▼UMB 設定のコツ

/M は「/M D0」などのように、アドレスの上 2 桁を記述します。/M は 16KB 単位で指定する点を理解すると、このオプションの使い方が見えてきます。たとえば D000h~DFFFFh の 64KB を UMB に設定するには、これを 16KB ずつに区切ります。

```
D0000h~D3FFFh ←── 1番目の領域（16KB）
D4000h~D7FFFh ←── 2番目の領域（16KB）
D8000h~DBFFFh ←── 3番目の領域（16KB）
DC000h~DFFFFh ←── 4番目の領域（16KB）
```

したがって/M への記述は「/M D0, D4, D8, DC」となります。

▼/NC オプションの使用

アプリケーションによっては/NC オプションを設定すると、アプリケーション内部のデータエラーが発生する場合があります。この場合は/NC オプションは取り外してください。

HYPERDSK.EXE *p.246*　EXDISK.EXE *p.244*

EXDISK.EXE

●RAMディスクを設定する

MELMM.386で確保したXMSメモリ上にRAMディスクに設定します。CONFIG.SYSで登録する方法と、コマンドラインから実行する方法の2通りがあります。

書　式

Ⅰ. **DEVICE=[d:][path]EXDISK.EXE　[E | X]　Size　/[DIR][W]**
Ⅱ. **EXDISK DEVICE　[E | X]　Size　/[DIR][W]**
Ⅲ. **EXDISK　[R][V][M nnnn][C]**

※Ⅰはデバイスドライバ、Ⅱ/Ⅲはコマンドラインでの書式

パラメータ	設定内容
E	EMS上にRAMディスクを確保する
X	EMB上にRAMディスクを確保する（既定値）
Size	RAMディスクとして確保する容量をKB単位で指定する
DIR	ディレクトリ数を32~480の範囲で指定する（既定値は128）
W	Windows上で容量の変更を可能にする
R	コマンドから作成したRAMディスクを取り消し、メモリを解放する
V	RAMディスクの状態を表示する
M nnnn	RAMディスクの容量を変更する
C	RAMディスクの内容をクリアする

使用例

●EMBに2MB（2048KB）のRAMディスクを作成する

```
DEVICE=EXDISK.EXE X 2048  ←――― CONFIG.SYSで指定する方法
EXDISK DEVICE=X 2048 ↵    ←――― コマンドラインから実行する方法
```

●作成可能なディレクトリ数を256個に設定する

```
DEVICE=EXDISK.EXE 2048/256
```

● Windows から RAM ディスク容量を調節できるようにする

```
DEVICE=EXDISK.EXE 2048 W
```

● RAM ディスクをコントロールする（コマンドラインから実行）

```
EXDISK R↵        ←―― RAM ディスクを解放する（取り去る）
EXDISK V↵        ←―― RAM ディスク情報を表示する
EXDISK M1024↵    ←―― RAM ディスク容量を 1024KB に変更する
EXDISK C↵        ←―― RAM ディスクの内容を消去する
```

解　説

▼コマンドラインからの実行

EXDISK.EXE は、コマンドラインから RAM ディスクを作成することもできます。また、コマンドラインから作成した RAM ディスクは、R オプションで RAM ディスクそのものを消してしまうことができます（CONFIG.SYS で作成した RAM ディスクは不可）。なお、コマンドラインから RAM ドライブを作成するには、あらかじめ CONFIG.SYS に「LASTDRIVE」コマンドで最終ドライブを登録しておく必要があります。

▼XMSDISK.EXE との違い

MELWARE には、RAM ディスクとして「EXDISK.EXE」と「XMSDISK.EXE」が付属しています。しかしアクセススピードが高速なこと、Windows 上（デバイスドライバモニタ）から容量を変更できること、などの理由から EXDISK.EXE の方が高機能です。EXDISK.EXE は「MELWARE for Windows」のディスクに含まれます。

▼Windows からの容量変更

「MELWARE for Windows」をインストールすると、Windows にグループ「MELWARE for Windows」が自動生成されます。これは Windows 上で使う各種ユーティリティが格納されていて、この中の「デバイスドライバモニタ」で RAM ディスク容量を変更することができます。ただし容量変更は RAM ディスク作成時に W オプションを付けている場合のみ有効です。

MELMM.386 *p.242*

HYPERDSK.EXE

●キャッシュディスクを設定する

MELMM.386で確保したXMSメモリ上にキャッシュディスクを設定します。CONFIG.SYSで登録する方法と、コマンドラインから実行する方法の2通りがあります。

書　式

Ⅰ. **DEVICE=[d:][path]HYPERDSK.EXE　[C:nnnn][CW:nnnn]**
[S | W][E][D]

Ⅱ. **HYPERDSK=[C:nnnn][CW:nnnn][S | W][E | D]**

※Ⅰはデバイスドライバ、Ⅱはコマンドラインでの書式

パラメータ	設定内容
C	MS-DOS動作時のキャッシュディスク容量をKB単位で指定する
CW	Windows動作時のキャッシュディスク容量をKB単位で指定する
S	書込処理をライトキャッシュバック（遅延書込）に設定する
W	書込処理をリアルタイムに行う（既定値）
E	HYPERDSKの機能を有効にする（既定値）
D	HYPERDSKの機能を一時的に無効にする

使用例

●DOS用に3072KBのキャッシュを割り当てる

DEVICE=HYPERDSK.EXE C:3072 ←── CONFIG.SYSで指定する方法

HYPERDSK C:3072 ⏎ ←── コマンドラインから実行する方法

●Windows起動時にキャッシュ容量を2048KBにする

DEVICE=HYPERDSK.EXE C:3072 CW:2048

●書込処理にもキャッシュを効かせる

DEVICE=HYPERDSK.EXE C:3072 CW:2048 S

解　説

▼XMSCHACHE.EXEとの違い

MELWAREにはディスクキャッシュとして「HYPERDSK.EXE」と「XMSCACHE.EXE」が付属しています。しかしアクセススピードが高速なこと、Windows上（デバイスドライバモニタ）から容量を変更できること、などの理由からHYPERDSK.EXEの方が高機能です。HYPERDSK.EXEは「MELWARE for Windows」のディスクに含まれます。

MELMM.386　*p.242*

ユーエムビーロード・シス　MELWARE for Windows

UMBLOAD.SYS

●UMBに常駐プログラムをロードする

デバイスドライバやTSR（常駐プログラム）をUMBにロードします。コマンドラインからの実行とCONFIG.SYSへの登録との2つの方法があります。

書　式

Ⅰ．**DEVICE=[d:][path]UMBLOAD.SYS　[/UA:n]　デバイスドライバ名**
Ⅱ．**UMBLOAD　[AU:n]　TSR名**

※Ⅰ はCONFIG.SYSでの書式、Ⅱ はコマンドラインでの書式

パラメータ	設定内容
/UA:n	指定するUMBブロック番号（n）にロードする

使用例　*p.155*

●PRINT.SYSを1番目のUMBブロックにロードする（CONFIG.SYS）

```
DEVICE=A:¥MEL4WIN¥UMBLOAD.SYS /UA:1 A:¥DOS¥PRINT.SYS
```

●DOSKEYコマンドを4番目のUMBブロックにロードする

```
UMBLOAD /UA:4 DOSKEY /INSERT ↵
```

●UMBのブロック番号/UMBの使用状況を調べる

```
UMBLOAD ↵
```

解　説

▼UMBLOADの特徴

UMBLOADはMS-DOSVer.5.0Aに付属するDEVICEHIGH（CONFIG.SYSコマンド）とLOADHIGH（MS-DOSコマンド）の機能を合わせ持ったコマンドです。UMBLOADの優れている点は、ロードするUMBブロックを指定できる点にあります。UMBのブロック番号を調べるには、「UMBLOAD ↵」と入力します。左端の「No」に表示される番号がUMBブロック番号です。なお、UMBLOADはCONFIG.SYSで使用する際には、UMBLOAD.SYSを使用し、コマンドラインから使用する際にはUMBLOAD.COMを使用します。書式は同じです。

DEVICEHIGH *p.189*

第4章
トラブル対策一覧

ここでは、CONFIG.SYS や AUTOEXEC.BAT をカスタマイズする際に突き当たる、さまざまなトラブルを回避する方法を紹介します

CONFIG.SYS
AUTOEXEC.BAT

CONFIG.SYSの設定が反映されない!!

この場合は次の2点をチェックしてください。

再起動を実行したか?

CONFIG.SYSを修正しただけでは、システムには何の影響もありません。修正内容を有効にするには、CONFIG.SYSを保存した後にリセットボタンを押して、パソコンを再起動してください。

CONFIG.SYSの格納場所は正しいか?

MS-DOSは起動ドライブのルートディレクトリにあるCONFIG.SYSを読み込み、システムに反映させます。それ以外の場所にCONFIG.SYSを作成しても無視されます。

起動時に「CONFIG.SYSが大きすぎます」と表示される

このメッセージは主に次のケースで表示されます。

デバイスドライバのロード失敗

何らかの原因でデバイスドライバをメモリにロードできなかった場合に、まれに表示されます。筆者の経験したケースでは、CONFIG.SYSに「DEVICEHIGH=SMARTDRV.SYS 2048 128」という行があるために、この症状が出ました。これは「DEVICE=SMARTDRV.SYS 2048 128」と修正することで回避できました。

通常はUMBにロードできなくても、自動的にコンベンショナルメモリにロードされるはずです。しかし、環境によっては、デバイスドライバとの相性が悪い場合に、このようなケースが起こり得ます。

実際にCONFIG.SYSの行数が多すぎる

起動時にMS-DOSはCONFIG.SYSをメモリに読み込んで、1行1行を解析します。しかしCONFIG.SYS自体のファイル容量が大きすぎると、MS-DOSが読み込めずにこのようなメッセージが表示されます。

「CONFIG.SYSに無効なコマンドかパラメータがあります」と表示される

CONFIGコマンドの記述をミスタイプしていると表示されます。表示された行番号のCONFIGコマンドとオプションの記述を修正してください。

「以下のファイルが無効または見つかりません」と表示される

このエラーメッセージは、CONFIG.SYSで記述してある場所に、指定したデバイスドライバが本当に存在しないときに表示されます。たとえば次のように記述したとします。

```
DEVICE=A:¥DOS¥KKCFUNC.SYS
```

このときAドライブのサブディレクトリ「DOS」に、デバイスドイライバファイル「KKCFUNC.SYS」が存在しないとエラーなります。また、パス設定やファイル名をミスタイプしている可能性もあるので、スペルのチェックが必要です。

起動の途中でパソコンがハングアップする

これはCONFIG.SYSまたはAUTOEXEC.BATの中に、メモリの重複使用などの不正な設定があるために起こります。たとえば、ハードディスクが使用しているアドレスを強制的にUMBに割り振ったときなどに起こります。

このような事態に陥ったら、次の手順で回避してください。

①別のドライブからのMS-DOS起動

MS-DOS起動用フロッピーディスク（または別のハードドディスクドライブ）からMS-DOSを起動します。

※このような事態に備えて、エディタ（SEDIT.EXE）やファイル修復コマンド（UNDELETE.EXE/UNFORMAT.COM/CHKDSK.EXEなど）を格納したMS-DOS起動用ディスクをあらかじめ用意しておくと安全です

②CONFIG.SYSの修正

ハングアップしたドライブのCONFIG.SYSをエディタで修正し、再起動します。

なお、ここで重要になるのが、どこが問題を引き起こしている行かを見分けるコツです。一般的な対策としては、CONFIG.SYSの中からあやしげなドライバを1行選んでREMコマンドで無効にして、再起動してみることです。もし正常に起動できたら、REM指定した行に原因があることがわかります。同じ要領で次々に行うと、問題のCONFIG.SYS行を絞り込むことができます。

? 起動時に「環境のためのメモリが足りません」と表示される

これは環境変数エリアが容量不足であるため、SETコマンドまたはPATHコマンドなどが正常に行われなかったときに表示されるメッセージです。この場合は、次のようにCOMMAND.COMに/Eオプションを付けて、容量を既定値256バイトより多く指定します。 参照 *p.102*

```
SHELL=A:¥COMMAND.COM /P /E:384   ←―― 384バイトに指定
```

? AUTOEXEC.BATは作ってあるのに実行されない!

この場合は次の3点をチェックしてください。

ファイル名は正しいか?

ファイル名が「AUTOEXEC.BAT」であるかどうかをもう一度チェックしてください。1文字でもスペルを間違っていると自動実行されません。

AUTOEXEC.BATの格納場所は正しいか?

MS-DOSは起動時に、起動ドライブのルートディレクトリにAUTOEXEC.BATを探します。それ以外の場所にあると無視されます。AUTOEXEC.BATを発見できないとDATEコマンドとTIMEコマンドを実行して、MS-DOSプロンプト(A>)を表示します。

COMMAND.COMに/Dオプションが付いていないか?

CONFIG.SYSのSHELL行に、「SHELL=¥COMMAND.COM /P /D」のように/Dオプションが付いているとAUTOEXEC.BATが自動実行されなくなります。/Dはマニュアルにない隠しオプションで、一時的にAUTOEXEC.BATを無視します。/Dは意図的にAUTOEXEC.BATを無効にしたいときに使うと便利です。

AUTOEXEC.BATで指定したドライブにテンポラリファイルが作成されない

AUTOEXEC.BATの中で、SET命令で環境変数「TEMP」にRAMドライブを指定すると、MS-DOSコマンドやDOSSHELLの動作を高速に実行できます。このときドライブ名に¥記号を付けないと、指定したドライブにテンポラリファイルが作成されずに、起動ドライブのルートディレクトリに作成されます。

```
SET TEMP = D:    ←――― ×
SET TEMP = D:¥   ←――― ○
```

JW2対応製品（一太郎・三四郎・花子・五郎）

インストール時に「メモリが足りません」と表示される

メインメモリの空き容量が小さくて、インストールプログラム「INST.COM」が起動するだけのメモリを確保できないときに表示されます。CONFIG.SYSやAUTOEXEC.BATからメモリを浪費しているプログラムを取り除いてリセットしてから、再実行してください。

インストール時に「ディスクの空き容量が足りません」と表示される

インストールに必要とするディスクの空き容量が、インストール先のハードディスクにない場合に表示されます。まず必要な容量を確保した上で、インストールコマンド「INST」を再実行してください。

? インストール時に「80286 以上の CPU が必要です」と表示される

使用しているパソコンの CPU（中央演算装置）が i80286（インテル製 CPU の型番）より下位の CPU を積んでいると、JW2 対応製品は使用できません。PC-9801VM や UV などがそれに当たります。これらの製品を快適に動かすには i80486 以上（98 MATE など）が理想的です。

? インストール時に「MS-DOS Ver.3.10 以上を使用してください」と表示される

MS-DOS Ver.3.10 より古いバージョンでは、JW2 対応製品は使用できません。できれば MS-DOS Ver.5.0A 以上をお使いください。

? 起動時に「EMS メモリが足りません」と表示される

JW2 対応製品は起動時に EMS メモリを発見すると、そこに作業領域を確保します。しかし、必要な容量を EMS に確保できないと上記のようなエラーメッセージを表示し、起動を中止します。

? 起動時に「ATOK8 が組み込まれていません」と表示される

JW2 対応製品を起動するには ATOK8 が使用できる状態になっていることが必要です。CONFIG.SYS に ATOK8 のデバイスドライバを登録するか、ADDDRV コマンド（***p.105***）で ATOK8 を登録してください。CONFIG.SYS で複数の FEP を登録している場合は、SELLKKC コマンド（***p.103***）で ATOK8 を選択してください。

「他のプログラムがプロテクトモードを使用しています」と表示される

JW2 対応製品を起動するときに、VCPI 規格に対応していないメモリマネージャを使っていると、このメッセージが表示されます。たとえば EMM386.EXE（NEC 版 MS-DOS）がそれに該当します。JW2 対応製品に付属するメモリマネージャ（EMS386.SYS）か、VCPI に対応したメモリマネージャ（MELWARE for Windows や MemoryServer II）を使用してください。 参照 *p.169*

起動時に「NECPM.MNG が見つかりません」と表示される 参照 *p.167*

環境変数「JW2P」に必要なパスが指定してないとこのメッセージが表示されます。環境変数は AUTOEXEC.BAT の中で SET 命令で指定します。

```
SET JW2P=A:¥JW2;A:¥TARO5                                        ←―― 一太郎 Ver.5 のみのパス
SET JW2P=A:¥JW2;A:¥SANSIRO                                      ←―― 三四郎のみのパス
SET JW2P=A:¥JW2;A:¥HANA3;A:¥JEDIT                               ←―― 花子 Ver.3 のみのパス
SET JW2P=A:¥JW2;A:¥GORO;A:¥JEDIT                                ←―― 五郎のみのパス
SET JW2P=A:¥JW2;A:¥TARO5;A:¥SANSIRO;A:¥HANA3;A:¥GORO;A:¥JEDIT   ←―― 複合
```

JW2 対応製品を起動できない

次の点をそれぞれチェックしてください。

- CONFIG.SYS に ATOK8 は登録されていますか？
- AUTOEXEC.BAT の PATH コマンドに「A:¥JUST5」が登録されていますか？
- AUTOEXEC.BAT で環境変数「JW2P」に必要なパスは指定していますか？
- メインメモリに 200KB（キロバイト）以上の空き容量がありますか？
- VCPI 規格対応のメモリ設定ドライバを CONFIG.SYS に登録してありますか？
- プロテクトメモリ（拡張メモリ）は 3072KB 以上使用可能ですか？
- CONFIG.SYS の FILES は 30 以上に設定してありますか？
- MS-DOS のバージョンは 3.10 以上を使用していますか？

パソコン起動時に変なメッセージが表示される

JW2対応製品に付属するメモリマネージャ「EMS386.SYS」は、パソコン起動時に次のようなエラーメッセージを表示することがあります。

「拡張メモリが利用できません」

プロテクトメモリを認識できないときに表示される重大なメッセージです。パソコンにプロテクトメモリが増設されていないか、増設したメモリボードのアドレスが間違っているか、またはメモリボードのハード的な故障であるかが考えられます。

「すでに拡張メモリマネージャが組み込まれています」

複数のメモリマネージャがCONFIG.SYSに重複して登録されている場合に表示されます。CONFIG.SYSに登録されているデバイスドライバを再チェックしてください。

「ページフレームを設定することができません」

EMSページフレームを確保するべきUMA領域のアドレス(既定値ではC0000h~CFFFFhの64KB)を別の周辺機器が使用していて、正常に確保できないときに表示されます。多くの場合、パソコンのサウンドBIOSが「使用する」に設定されていることに原因があります。HELPキーを押しながらパソコンを起動し、サウンドBIOSを「切り離す」に設定してください。

「スワップファイルが読込専用のため使用できません」と表示される

一太郎Ver.5は、起動時にJW環境設定で指定してあるパス(既定値はA:¥JW2)にスワップファイル(JW_SWAP.$$$)を発見するとこのメッセージを表示します。通常は一太郎を正常終了するとスワップファイルは自動削除されますが、何らかの原因で過去に一太郎Ver.5を起動したまま電源をOFF(またはリセット)にしたことがあると、このスワップファイルが残ったままになります。この場合は次のように入力すると、このエラーメッセージを消せます。

```
ATTRIB A:¥JW2¥JW_SWAP.$$$ -R ↵
```

「編集作業領域〈ファイル〉が読込専用のため使用できません」と表示される

一太郎 Ver.5 は起動時に JW 環境設定で指定してあるパス(既定値は A:¥JW2)に編集作業領域ファイル(JW_WMM.$$$)を発見するとこのメッセージを表示します。通常は一太郎を正常終了すると編集作業領域ファイルは自動削除されますが、何らかの原因で過去に一太郎 Ver.5 を起動したまま電源を OFF(またはリセット)にしたことがあると、このファイルが残ったままになります。この場合は次のように入力すると、このエラーメッセージを消せます。

ATTRIB A:¥JW2¥JW_WMM.$$$ -R ⏎

JW2 対応製品の動作が遅くて困る

この場合、次の点をチェックしてください。

編集作業領域のメモリサイズは適切ですか　p.171

プロテクトメモリから編集作業領域のメモリサイズを差し引いた容量が 2MB 以上になるように設定してください。逆に編集作業領域が少なすぎると、「編集作業領域」をハードディスク上に編集領域を確保しようとして、この場合も動作は遅くなります。

プロテクトメモリは十分ですか

JW2 対応製品では 3MB~4MB 以上のプロテクトメモリを動作条件としていますが、快適に動かすには 8MB~14MB クラスが理想的です。特に五郎は多くのメモリを消費するので、他の JW2 対応製品をスワップファイルを使わずにすべて同時起動するには、14MB 前後のプロテクトメモリが必要になります。

なお、同じプロテクトメモリでもパソコン背面の拡張スロットに装着するタイプメモリボードを使用している場合は、メモリ容量が十分であってもアクセスは遅くなります。32 ビット幅でデータ送信ができる内部増設型メモリボードを使用することをお勧めします。　p.67

CPU は i80486 以上ですか

使用するパソコンの CPU が i80286/i80386SX クラスだと、検索処理や画像表示などで動作が鈍くなります。JW2 対応製品はグラフィック処理が多いので、i80486 以上のマシンの使用が理想的です。

✿フォントキャッシュは設定していますか　参照 p.173

参照 p.173

一太郎 Ver.5 ではイメージ編集モードでフォントイメージを表示するときに、フォントファイルへのアクセスがひんぱんに行われます。フォントキャッシュを設定することで、アクセス回数を減らし、高速な画面表示ができるようになります。

✿ディスクキャッシュは設定していますか

ディスクキャッシュを設定すると辞書ファイルへのアクセスや、コマンド実行が高速になる場合があります。JW2 対応製品には EMSCACHE.SYS というディスクキャッシュドライバが付属しています。

DPMI コマンドを使うと動作が不安定になる

MS-DOS Ver.5.0A 以上に付属する DPMI コマンドを実行する場合は、ATOK8A.SYS に/Z=1 オプションを必ず付加してください。また、DPMI コマンドを使用しないのに/Z=1 オプションを付けておくと、これも JW2 対応製品の動作が不安定になるので注意してください。

JS TextEditor を起動できない

AUTOEXEC.BAT に「SET JW2P=A:¥JEDIT」が登録されていない可能性があります。ない場合はテキストエディタ（MS-DOS の場合は SEDIT コマンド）で登録してください。

Windows

MS-DOS Ver.5.0AとWindows3.1のメモリマネージャは同じなの?

MS-DOS Ver.5.0AとWindows3.1には、双方に同じファイル名のメモリマネージャ(HIMEM.SYS/EMM386.EXE)が入っています。MS-DOSアプリケーションを動かす上では、双方とも大きな差異はありません。しかし、Windows3.1を動かす際には、必ずWindows3.1に付属するメモリマネージャを使用してください。Windows3.1に付属するHIMEM.SYSはXMS規格のバージョンがXMS3.0で、MS-DOSの方はXMS2.0です。新98MATEなど14.6MB以上のプロテクトメモリを扱うには、XMS3.0以上が要求されます。また、MS-DOS Ver.5.0Aに付属するディスクキャッシュSMARTDRV.SYSは、Windows3.1では機能しない場合があります。

「一般保護違反」「ページ違反」などのエラーが表示される!

このメッセージは、メモリ管理上のトラブルが発生し、Windowsを正常に動作できなくなったときに表示されます。このエラーは、明確な原因を特定しにくいという一面があります。一般的には、システムリソース不足が原因と思われるケースが多いようです。システムリソースとはWindowsの各種動作を管理する64KBの領域と、グラフィック処理の情報を管理する64KBのメモリ領域です。これらはアプリケーションを複数起動したり、ウィンドウ(ダイアログボックス)をたくさん開いたりすると、メモリよりもリソースの方が先に容量不足になってしまうことがあります。

この他に、メモリマネージャとWindowsの相性が悪いと表示されるケースもあります。この場合は、別のメーカーのメモリマネージャを使ってトライしてください。

p.159

Windowsのメモリを節約するにはどうすればいいの?

Windowsでできるだけたくさんの空きメモリを使用できるようにするには、次の3点をチェックしてください。

✿CONFIG.SYSとAUTOEXEC.BATの中身を調べる

CONFIG.SYSでEMSメモリを個別に確保していたり、日本語入力プログラム(FEP)、マウスドライバなどのデバイスドライバを登録していると、Windowsで使用できるメモリが少なくなります。また、AUTOEXEC.BATにTSR(常駐型プログラム)がある場合も、メモリを消費しています。

✿色数の多い壁紙を使っているかどうかを調べる

フルカラーや256色などの色数の多い壁紙、元の面積の大きい壁紙を使うと消費メモリは多くなります。壁紙をまったく使わないと500KB~800KBほど節約できます。

✿フォントドライバやIMEを統一する

複数のフォントドライバ(TrueType、FontWave、JSフォントなど)やIME(ATOK8、MS IME、AIかな漢字など)がWindowsに登録してあると、システムリソースが余分に使われます。これらはできれば統一して、他はコントロールパネルから削除するようにすると、メモリとシステムリソースを節約できます。

? Windows3.1が起動できなくなった!

この場合は次の3点をチェックしてください。

✿XMS2.0以上に対応したメモリマネージャを使っているか

XMS2.0以上に対応していないメモリマネージャ(JW2対応製品に付属するEMS386.SYSなど)をCONFIG.SYSに登録していると、Windows3.1が起動できなくなります。Windows3.1はXMSからメモリ供給を受けて、それをDPMI規格(バージョン0.9)のメモリとして使用します。

✿プロテクトメモリは4MB以上か

Windowsは4MB(4096KB)以上のプロテクトメモリがないと起動できません。

✿パス設定は正しいか

AUTOEXEC.BATのPATH設定で、「A:¥WINDOWS」が指定されていないと、Windowsは起動しません(例: PATH A:;A:¥DOS;A:¥WINDOWS)。

Windows 起動時に「HIMEM.SYS が無効です」と表示される

メモリマネージャが XMS 規格 Ver.2.0 以上に対応していない環境で、Windows3.1 を起動しようとすると、「HIMEM.SYS が無効です。Windows のディスクから HIMEM.SYS をコピーしてください」と表示されます。主に JW2 対応製品（一太郎 Ver.5/三四郎/花子 Ver.3/五郎など）をインストールした後に Windows を起動するとこのメッセージが出ます。これは CONFIG.SYS に自動登録される EMS386.SYS が XMS 規格に対応していないためです。

Windows の起動が極端に遅くなってしまった!!

この場合は次の 2 点をチェックしてください。

キャッシュは効いているか

ディスクキャッシュの中には、Windows 環境では機能しないものがまれにあります。たとえば MS-DOS Ver.5.0A に付属する SMARTDRV.SYS がそれです。この場合は Windows3.1 に付属する HIMEM.SYS/EMM386.EXE/SMARTDRV.EXE を使用してください

古いタイプのハードディスクを使っていないか

初期の SCSI 型ハードディスクや SASI 型ハードディスクは、データの読み取り/転送速度が極めて遅いため、Windows の起動は 3~5 分もかかる場合があります。なお、Windows3.1 では一部の SCSI 型ハードディスク用に若干のスピードアップをはかる機能(ダブルバッファ)があるので、必要に応じてCONFIG.SYSにSMARTDRV.EXE を登録してください。 参照 p.162

14.6MB 以上のプロテクトメモリが活用されていない!!

MS-DOS Ver.5.0A に付属する HIMEM.SYS(XMS2.0 対応)や、初期の Windows 3.1 に付属する HIMEM.SYS（XMS3.0 対応）は、14.6MB 以上のプロテクトメモリを認識できません。新 98MATE などの 14.6MB を超えるプロテクトメモリを認識させるには、XMS3.0 以上に対応したメモリマネージャ（MELWARE for Windows や MemoryServer II)を使う必要があります。Windows3.1 に付属する HIMEM.SYS

を利用するには、1994年2月以降に出荷された製品を使うか、またはアップグレードサービスを受けて最新のHIMEM.SYSを使ってください。

MS-DOS

MS-DOSにリブートコマンドはないの？

参照 p.99、101

MS-DOS Ver.5.0Aには「CHGEV」というリブートコマンドがあります。ただしCHGEVは製品マニュアルでは機能やオプションは説明されていない隠しコマンドです。したがってNECユーザーサポートセンターでのサポートは受けられませんので、使用には十分注意を払ってください。またCHGEVはリセットボタンを押したのと同じリブートしかできません。高速リブートを行うには、HSBやメニューソフトのリブートコマンドを使用してください。

MS-DOS Ver.5.0AをVCPI/DPMI規格対応にさせる方法はあるの？

NEC版MS-DOS Ver.5.0AはVCPI規格に対応していません。VCPI規格に対応させる方法としては、SYMDEBコマンドでEMM386.EXEを修正する方法が雑誌などで紹介されていますが、動作が不安定になるなどの障害があり、実用向きではありません。むしろVCPI規格に正式対応しているMELWARE for WindowsやMemoryServer IIを使った方が安全で確実です。

また、MS-DOS Ver.5.0AをDPMI規格に対応させるには、MS-DOS起動後に「DPMI ↵ 」と入力してください。これでDPMIサーバが起動します。MS-DOS Ver.5.0にはDPMIコマンドは含まれていないので、今のところは対処方法がありません。MemoryServer IIをお使いの場合は、コマンドラインから「DPMI32 ↵ 」を実行すると、DPMIサーバを起動できます。MELWARE for WindowsはDPMIには対応していません。

DOSSHELLを使うとメモリはあるのにメモリ不足になる？

参照 p.136

DOSSHELLは起動時に空いているプロテクトメモリをすべてXMSメモリとして確保します。したがってアプリケーションの要求に応じてプロテクトメモリをXMSやEMSに配分するVMM386.EXE(MemoryServer II)やMELEMM.386(MELWARE for Windows)を使用している場合は、起動時にEMSやVCPI用のメモリを必要とするアプリケーション(一太郎Ver.5や桐など)は起動不可能になります。これを防ぐにはVMM386.EXEには/Xオプション(***p.229***)、MELEMM.386には/XMSオプション(***p.242***)で、それぞれXMSとして確保できる容量を制限してください。

なお、EMM386.EXE(MS-DOS Ver.5.0付属)やEMS386.SYS(一太郎Ver.5付属)は、あらかじめ一定のEMSメモリを確保したまま起動します。メモリ配分がフレキシブルでないという不器用さが、逆にDOSSHELLとの相性をよくしています。

/MOVEHDBIOSが効かない！

参照 p.201

/MOVEHDBIOSは、SCSI型またはSASI型ハードディスクのBIOS ROMを移動します。ただし、IDE型ハードディスクと、98NOTEの内蔵ハードディスク(SASI型)のBIOS ROMは移動できません。98NOTEの内蔵ハードディスクはSASIをエミュレートしていますが、ハードウェアはIDE型を使っているためです。

したがって、98MATEや98FELLOWなどの内蔵型ハードディスク(IDE)を接続している機種では、/MOVEHDBIOSは指定してもあまり効果がありません。

/MOVEHDBIOSでBIOS ROMを移動したのに4KBしか空かない

参照 p.88

EMM386.EXEに/MOVEHDBIOSを付けると、SCSI型ハードディスクのBIOS ROM(DC000h~DCFFFhまでの4KB)が、UMA領域の未使用領域(A5000h以降)に移動します。ただし、SCSI型のROMは起動時にはDC000h~DDFFFhまでの8KBを占有しており、/MOVEHDBIOSを指定しただけでは、DD000h~DDFFFhまでの4KB(未使用領域)は中途半端に残ってしまいます。この場合は、次のように/Iオプションでこの領域もUMBに指定すると、連続したUMBにすることができます。

```
DEVICE=A:¥DOS¥EMM386.EXE /P=256 /UMB /MOVEHDBIOS /I=DC00-DFFF
```

MemoryServer II

? E8000hをUMBに設定できない!

VMM386.EXEに/NECID (Windows起動時のIDチェック対策) を付けた状態で、E8000h~E8FFFh (N88-BASICの開始アドレス) までをUMBに設定 (/Uオプション) すると、パソコン起動時 (またはWindows起動時) にハングアップします。OPTUMBでも「Windowsを使用する」を選択すると、E8000h~E8FFFhは選択できないようになっています。ちなみにMELWARE for WindowsではE8000h~E8FFFhをUMBに設定しても、問題なくWindowsを起動できます。

MELWARE for Windows

? SCSI型ハードディスクのBIOS ROMを移動できない

MELEMM.386に/SW1オプションを付けると、SCSI型またはSASI型のハードディスクのBIOS ROMを移動できます (IDE型のBIOS ROMは移動できない)。ただし、機器環境によってはSCSI型のBIOS ROMが移動できない場合があります。筆者が確認したケースでは、98MATEにIDEとSCSIを接続した環境で、この現象が見られました。この場合は他のメモリマネージャを使うとよいでしょう。

? Newton-98が起動できない

MELWARE (MELEMM.386) を設定してあるCONFIG.SYS環境でNewton-98 (Admiral System) が起動できない場合は、他のメモリマネージャを使ってください。環境によっては、相性が悪くてハングアップするなど症状が出ます。

その他

? どうしてUMBはD0000hでなくてD0010hから始まるの?

VMAP/MEMなどでメモリマップを表示すると、計算上ではUMBになっているはずのエリアより10h（16バイト）後からUMBが始まっていることに気が付きます。たとえばD0000h~DFFFFhまでをUMBに指定した場合、メモリマップ上ではD0010hからUMBが始まっています。実は各UMBブロックの先頭には、必ずこの10h（16バイト）分のエリアが確保されているのです。これは「メモリコントロールブロック（MCB）」と呼ばれる領域で、この中にはUMBのサイズや使われ方の情報が格納されています。MS-DOSはこのMCBをたよりにUMBをアクセスしています。メモリマップではMCBの表記を省略しているものが多いようです。

? 16進数は読みづらいので10進数に変換したい

p.55

メモリマップユーティリティの多くはアドレスやサイズは16進数で表示します。16進数を10進数に変換するには、三四郎やLotus1-2-3のDECIMAL関数を使うと簡単です。たとえばD0000hを10進数に変換するには、「DECIMAL（"D0000"）」と入力します。

? ノート型パソコンで連続したUMBが確保できない

98NOTEでは、D8000h~DBFFFhあたりに内蔵ハードディスク（SASI）とRAMドライブのBIOS ROMがあります。このため、ディスクトップパソコンでは連続したUMBになるこの領域はUMBが途切れてしまいます。この領域はUMBにする方法はありません。主なノート型パソコンのUMBにできないアドレスは次の通りです。

98NOTE シリーズ	：D8000h~DBFFFh
PC-386NOTE A/BookL	：D4000h~DFFFFh
PC-386NOTE AE/AR/W/WR	：D8000h~DBFFFh
PC-486NOTE AS	：D8000h~DBFFFh

DiskX IIのデバイスドライバがUMBにロードできない!

DiskX IIのデバイスドライバ「DISKX.SYS」は、圧縮ドライブのクラスタサイズによって、消費されるメモリサイズが変化します。クラスタサイズ4KBでは約45KBを消費し、クラスタサイズ16KBでは約65KBのメモリを消費します。このように環境によってDISKX.SYSの消費メモリは異なるので、UMBブロックの状態によってはロードできないこともあり得ます。UMBにロードできない場合はコンベンショナルメモリにロードされます。コンベンショナルメモリに45KB~65KBクラスのデバイスドライバが居座ると、他への影響が甚大です。この場合は/EMSオプションを付けてEMSへロードさせた方がよいでしょう。

ノートンディスクドクター(NDD)が起動できなくなった

この場合は次の3点をチェックしてください。

ディスクキャッシュは影響していないか

MemoryServer IIに含まれるディスクキャッシュ「DC10.EXE」、またはMELWARE for Windowsに含まれる「HYPERDSK.EXE」をCONFIG.SYSから外します。

ATOK8は影響していないか

ATOK8をCONFIG.SYSから外します。NDDの販売元では、環境によってはATOK8と市販のメモリマネージャ(MELWAREとMemoryServer II)との相性が悪い場合があることを指摘しています。

DiskX IIは影響していないか

DISKX.SYSをCONFIG.SYSから外します。

この3点をチェックしても、ノートンディスクドクターを起動できない場合は、メモリマネージャを別の製品に変えてみる必要があるでしょう。

ハードディスクの起動メニューを表示させたい!

ハードディスクの起動メニューをいったん「自動起動」に設定すると、次回からは起動メニューを経由せずに、ダイレクトに指定ドライブから起動されます。このとき再び起動メニューを表示するには、次の2つの方法があります。

CHKENVコマンドの隠しオプションを使う

「CHGEV /M⏎」と入力するとハードディスクの起動メニューを表示できます。

TABキーを使う

リセットボタンを押しながらTABキーを押し続けると、ハードディスクの起動メニューを表示できます。

本書のトラブル対策どおりに対処してもうまくゆかない!

本書のトラブル対策一覧で記述する内容が、すべてのユーザーのケースに当てはまるとは限りません。使用するMS-DOSやアプリケーションのバージョン、パソコンや周辺機器によっては、異なる結果が出る場合もあるからです。

また、同じ機種のパソコンでも、出荷次期によりROMバージョンが違ってくる場合があります。同様に、MS-DOSやアプリケーションも、バージョン番号は同じであっても、その出荷時期によって内部が微妙に変更されている場合があります。

付　録

単位早見表

単　位	読　み	意　味
inch	インチ	1inch=2.51cm
Byte (B)	バイト	1Byte=8bit
bit	ビット	1bit は 2 進数 1 桁に相当
bps	ビーピーエス	1 秒間に送れるビット数
CPI	シーピーアイ	1inch 当たりの文字数
cps	シーピーエス	1 秒当たり印字文字数
dB	デシベル	音量を表わす単位
DPI	デーピーアイ	1inch のドット数（プリンタなどの解像度）
FLOPS	フロップス	1 秒当たり浮動小数点演算の回数を表わす
GB	ギガバイト	1GB=1024MB
Hz	ヘルツ	電波や音の 1 秒間の振動数を示す単位
KB	キロバイト	1KB=1024B
KHz	キロヘルツ	1KHz=1000Hz
LPI	エルピーアイ	1inch 当たりの走査線の本数を表わす
m	ミリ	1000 分の 1
MB	メガバイト	1MB=1024KB
MHz	メガヘルツ	100 万 Hz
MIPS	ミップス	1MIPS は 1 秒間に 100 万回の命令を実行
msec	ミリ秒	1000 分の 1 秒
n	ナノ	10 億分の 1
nm	ナノメーター	10 億分の 1m
nsec	ナノ秒	10 億分の 1 秒
Pt	ポイント	1Pt=0.3514cm（活字の大きさを表す単位）
sec	秒	時間の単位
TB	テラバイト	1TB=1024GB
TPI	ティーピーアイ	記録密度の単位
μ	マイクロ	100 万分の 1
μm	マイクロメーター	100 万分の 1m
μsec	マイクロ秒	100 万分の 1 秒

インフォメーションセンター連絡先

会社名	電話番号	FAX
アイ・オー・データ機器　(本社)	0762 (60) 3366	0762 (60) 3360
(東京)	03 (3254) 0301	03 (3254) 9055
アイシーエム　(東京)	03 (5294) 6571	03 (5294) 6575
(大阪)	06 (339) 9390	06 (339) 9380
アシスト	044 (951) 5678	03 (3437) 0758
アスキー	03 (5351) 8080	03 (5351) 8721
アップルコンピュータジャパン	03 (5562) 5600	
アドミラルシステム	048 (255) 8663	048 (250) 1284
アルゴ	06 (930) 2211	06 (930) 2213
アルプスシステムインテグレーション	045 (541) 0282	045 (541) 0284
インテルジャパン	0120 (1) 80387	
インターソフト	03 (3842) 6017	03 (3842) 6012
エー・アイ・ソフト　(本社)	0263 (33) 3632	0263 (33) 3052
(東京)	0424 (85) 7444	
エプソン販売　(パソコン)	0424 (99) 7122	
(プリンタ)	0424 (99) 7133	
カノープス電子	078 (992) 6830	078 (992) 2998
キャノン	03 (3455) 9320	
コーパス	03 (5543) 3683	03 (5543) 3677
コンパック	0120-101589	
サーキットリンケージ	0473 (56) 6342	0473 (56) 6700
サザンパシフィック	045 (314) 9514	045 (317) 1351
サムシンググッド	03 (3232) 0801	03 (3232) 0963
システムハウスミルキーウェイ	03 (5563) 0370	03 (5563) 0371
ジャストシステム	(東京)	
(一太郎・三四郎・五郎・花子)	03 (5470) 6001	03 (5470) 6030
(一太郎・三四郎・五郎・花子以外)	03 (5470) 6003	03 (5470) 6030
	(大阪)	
(一太郎)	06 (886) 6001	06 (390) 1666
(一太郎以外)	06 (886) 6003	06 (390) 1666

会社名	電話番号	FAX
ジャストシステム（徳島）	0886（55）6001	0886（55）6840
セイコーエプソン　（調布）	0424（99）7122	
（大阪）	06（397）0915	
ソリマチ情報センター	0258（32）6051	0258（37）1239
ダイナウェア	0727（27）2051	0727（27）2066
ツァイト	03（3299）0461	03（3299）0416
テグレット技術開発	03（3932）9685	03（3934）7471
デービーソフト	011（807）6700	011（807）6710
バックス	0427（24）9200	0427（28）6864
ピーシーエー	03（5211）2700	03（5211）2740
ボーランドジャパン	03（5350）9380	03（5350）9369
プロサイド	03（3255）1517	03（3255）1513
メルコ	052（619）1821	
マイクロソフト	03（5454）2300	03（5454）7951
マイクロデータ	03（3232）9801	
メガソフト	06（386）2043	06（386）2123
ロータス	03（5496）3101	03（5496）3690
沖電気工業	0120-296007	
東芝	03（3252）3100	
日本アイ・ビー・エム	0120-041992	
日本電気	03（3452）8000	
日立製作所	0120-258012	03（3298）1150
富士通	0120-009940	
緑電子	044（989）1441	044（989）8930
管理工学研究所	03（3405）1827	03（3475）5108
大塚商会	0473（29）0213	
日本ワードパーフェクト	03（3780）0451	

パソコン修理窓口（故障などの保守サービス）

会社名	〒	連絡先	電話番号
NEC			
C&C サービスセンター	154	東京都世田谷区桜新町 1-34-25	03-3705-8551
沖縄サービスセンター	901-21	沖縄県浦添市伊祖 2-7-11	0988-76-2255
関越サービスセンター	321	宇都宮市元今泉 5-9-13	0286-63-2384
九州北サービスセンター	814	福岡市早良区原 4-23-10	092-831-6421
四国サービスセンター	761	高松市三条町字中所 674-2	0878-66-2177
信州サービスセンター	390	長野県松本市大字出川町 1069-1	0263-27-4551
大阪サービスセンター	536	大阪市城東区関目 6-9-28	06-934-1346
中九州サービスセンター	862	熊本市大江町渡鹿 275-2	096-381-3322
中国西サービスセンター	731-01	広島市安佐南区西原 9-6-10	082-874-9700
中国東サービスセンター	700	岡山市野田 3-8-11	0862-43-8089
東北南サービスセンター	980	仙台市青葉区堤町 1-3-3	022-275-6641
東北北サービスセンター	020-01	盛岡市月ケ丘 3-49-50	0196-41-3218
北海道サービスセンター	060	札幌市中央区北 4 条西 16 丁目 1	011-621-1320
名古屋サービスセンター	461	名古屋市東区山田東町 2-65	052-722-3851
EPSON			
横浜サービスセンター	221	横浜市神奈川区鶴屋町 2-26-4 第三安田ビル	045-316-4820
金沢サービスセンター	920	金沢市西念町リ 32-2金沢M.Gビル	0762-62-3216
広島サービスセンター	730	広島市中区三川町 7-1 セイコー広島ビル	082-240-5411
札幌サービスセンター	060	札幌市中央区北 1 条西 2 丁目 札幌時計台ビル	011-222-2821
鹿児島サービスセンター	890	鹿児島市中央町 9-1 西鹿児島第一生命ビル	0992-54-6079
松本サービスセンター	390	長野県松本市中央 2-1-27	0263-36-7082
仙台サービスセンター	980	仙台市青葉区一番町 4-1-1 仙台セントラルビル	022-263-3691

会社名	〒	連絡先	電話番号
大阪サービスセンター	532	大阪市淀川区宮原 3-5-24 新大阪第一生命ビル	06-397-0930
東京サービスセンター	182	東京都調布市布田 1-31-1	0424-80-2891
福岡サービスセンター	812	福岡市博多区博多駅東 2-6-23　住友博多駅前第二ビル	092-471-0761
名古屋サービスセンター	460	名古屋市中区新栄町 2-13 栄第一生命ビル	052-962-7038

索　引

● あ

● か

● さ

● た

● な

● は

● ま

● ら・わ

■著者略歴

池田　龍之介（いけだ　りゅうのすけ）

1959年　新潟県柏崎市生まれ。

欧州留学中にバルセロナにてUFOに遭遇、天命を知る。

駒沢大学経済学部卒業後、商社に勤務。

システム開発マネージャ、経営企画室長を経て独立。

現在、C&R研究所代表取締役。

ロックグループ「Maria」バンドマスター。

「宇宙エネルギー研究会」主宰。

■主な著書

「入門　Z's STAFF KiD98 Ver3」「入門　KiT」

「MS-DOS Ver.5リファレンスガイド」「教育版　弥生販売管理3」

（以上、エーアイ出版）　「入門　JG Ver.3」（明日香出版）

「DiskX II Ver.2ハンドブック」「Lotus1-2-3印刷のすべてがわかる本」

「はじめて使う一太郎Ver.5　快適環境設定&基本操作マスター編」

「はじめて使う一太郎Ver.5　と・こ・と・ん活用編」「はじめて使う三四郎」

「Windows版　はじめて使う一太郎Ver.5　快適環境設定&基本操作マスター編」

「Windows版　はじめて使う一太郎Ver.5　と・こ・と・ん活用編」

「はじめて使う花子Ver.3」　（以上、ナツメ社）

CONFIG.SYSのすべてがわかる本

著　者　池田　龍之介
発行者　田村　正隆

発行所　株式会社ナツメ社
東京都千代田区神田神保町1-52 加州ビル2F（〒101）
電話　03（3291）1257（代表）　FAX　03（3291）5761
振替　00130-1-58661

制　作　ナツメ出版企画株式会社
東京都千代田区神田神保町1-52 加州ビル3F（〒101）
電話　03（3233）8961

印刷所　ラン印刷社

ISBN4-8163-1753-8　Printed in Japan

ISBN4-8163-1753-8 C2055 P1500E

ナツメ社■定価1,500円[1,456円]

パソコン環境をより快適にしたい！
だからメモリを増設したい、
メモリを有効に使いたい！
そんな98ユーザーに捧げる、
メモリ環境を決定づけるキー・ファイル
“CONFIG.SYS”設定のコツがわかる本。
[MS-DOS5.0A/Windows3.1対応]

■基礎知識編
CONFIG.SYSとは何か、メモリとは何かなど、パソコン環境を考える上で基本となる概念や仕組みを図解・解説しています。

■徹底活用編
MS-DOS5.0A、MemoryServerII、MELWARE for Windowsのそれぞれに付属するメモリ管理ドライバを使って、メモリの有効活用、高速アクセスなどの環境を実現する方法を徹底的に解説。また、Windows3.1、一太郎Ver.5、Lotus1-2-3 R2.4Jなど主要ソフトを快適に動かすための、CONFIG.SYSやメモリ環境についても解説しています。

■リファレンス編
CONFIG.SYSをカスタマイズする上で不可欠なコマンド、デバイスドライバの機能リファレンスです。